HARDWARE AND SOFTWARE IMPLEMENTATION OF A LOW POWER ATTITUDE CONTROL AND DETERMINATION SYSTEM FOR CUBESATS

By

Jesse Frey

RECOMMENDED:

Dr. Denise Thorsen

Dr. Dejan Raskovic

Dr. Joseph Hawkins
Advisory Committee Chair

Dr. Charles Mayer
Chair, Department of Electrical and Computer Engineering

APPROVED:

Dr. Douglas Goering
Dean, College of Engineering and Mines

Dr. John Eichelberger
Dean of the Graduate School

12/8/14
Date

HARDWARE AND SOFTWARE IMPLEMENTATION OF A LOW POWER ATTITUDE CONTROL AND DETERMINATION SYSTEM FOR CUBESATS

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Jesse Frey, B.S.

Fairbanks, Alaska

December 2014

Abstract

In recent years there has been a growing interest in smaller satellites. Smaller satellites are cheaper to build and launch than larger satellites. One form factor, the CubeSat, is especially popular with universities and is a 10 cm cube. Being smaller means that the mass and power budgets are tighter and as such new ways must be developed to cope with these constraints. Traditional attitude control systems often use reaction wheels with gas thrusters which present challenges on a CubeSat. Many CubeSats use magnetic attitude control which uses the Earth's magnetic field to torque the satellite into the proper orientation. Magnetic attitude control systems fall into two main categories: active and passive. Active control is often achieved by running current through a coil to produce a dipole moment, while passive control uses the dipole moment from permanent magnets that consume no power. This thesis describes a system that uses twelve hard magnetic torquers along with a magnetometer. The torquers only consume current when their dipole moment is flipped, thereby significantly reducing power requirements compared with traditional active control. The main focus of this thesis is on the design, testing and fabrication of CubeSat hardware and software in preparation for launch.

Table of Contents

Page

Signature Page . . . . . **i**

Title Page . . . . . **iii**

Abstract . . . . . **v**

Chapter Table of Contents . . . . . **vii**

List of Figures . . . . . **xi**

List of Tables . . . . . **xiii**

List of Appendices . . . . . **xv**

List of Acronyms . . . . . **xvii**

Acknowledgments . . . . . **xix**

Chapter 1 Introduction **1**

1.1 CubeSats 1

1.2 CubeSat ACDS 1

1.2.1 Passive Control 2

1.2.2 Active Control 2

1.2.3 Low Power Magnetic Torquer Control 2

1.3 Alaska Research CubeSat 2

1.4 Attitude Control and Determination System 3

Chapter 2 Background **5**

2.1 Low Power Magnetic Torquers 5

2.1.1 Hard Magnetic Core 5

2.1.2 Torquer Pair 6

2.1.3 Torquer Sets 7

2.2 Detumble Algorithm (Mode 1) 7

2.3 Alignment Algorithm 8

2.3.1 Mode 2 9

2.3.2 Mode 3 10

2.4 Simulation Results 10

2.5 Concept of Operations for the Alaska Research CubeSat 10

2.5.1 B-dot Control 12

Chapter 3 Hardware Implementation of the Attitude Control and Determination System (ACDS) **15**

3.1 Mechanical Configuration of the Alaska Research CubeSat (ARC) 15

Page

3.1.1 Alaska Research CubeSat Attitude Control and Determination System Hardware . . . 17
3.1.2 Bus Communication . . . 18
3.2 Components of the Attitude Control and Determination System . . . 18
3.3 Torquers . . . 20
3.3.1 Cores . . . 20
3.3.1.1 Core Sizing . . . 21
3.3.2 Coils . . . 22
3.3.3 Drivers . . . 22
3.3.3.1 Driving Waveform . . . 22
3.3.4 Torquer Diagnostics . . . 25
3.4 Sensors . . . 26
3.4.1 Magnetometer . . . 27
3.4.1.1 Magnetometer Amplifier . . . 27
3.4.1.2 Magnetometer Analog-to-Digital Converter . . . 28
3.4.1.3 Magnetometer Operation . . . 28
3.4.1.4 Magnetometer Calibration . . . 30
3.4.2 MEMS Gyros . . . 31
3.5 Embedded System . . . 31
3.5.1 Processor . . . 31
3.5.2 SD card . . . 31
3.5.3 ARC Bus Communication . . . 32
Chapter 4 Helmholtz Cage . . . 33
4.1 Hardware . . . 33
4.2 Software . . . 34
4.2.1 Compatibility . . . 35
4.3 Testing . . . 36
Chapter 5 Software . . . 39
5.1 Overview . . . 39
5.2 System Operations Overview . . . 39
5.2.1 Rotation Runaway . . . 40
5.3 Ground Station Commands . . . 40
5.4 Algorithm Software . . . 40
5.4.1 B-dot Algorithm . . . 40

Page
5.5 Auxiliary Software Functions . . . 41
5.5.1 Torquer Flipping Logic . . . 41
5.5.1.1 Status Tracking . . . 41
5.5.1.2 Torquer Feedback . . . 41
5.5.2 Torquer Compensation . . . 42
5.5.2.1 Compensation Data Set . . . 42
5.5.2.2 Compensation Routine . . . 42
5.5.3 Data Logging . . . 42
5.5.4 On-Board Data Processing . . . 43
5.5.5 Beacon Data . . . 43
Chapter 6 Verification . . . 45
6.1 Torquer Offsets and Repeatability . . . 45
6.1.1 Torquer Offsets . . . 45
6.1.2 Torquer Repeatability . . . 48
6.2 Magnetometer Verification . . . 48
6.2.1 Initial Verification . . . 51
6.2.2 Torquer Compensation . . . 51
6.2.3 Compensation Testing . . . 52
6.2.4 Compensation Data Transfer . . . 52
6.2.5 Embedded Compensation Testing . . . 55
6.3 B-dot Controller Simulations . . . 60
6.4 Detumble Flip Test . . . 62
Chapter 7 Conclusion and Future work . . . 65
7.1 Conclusion . . . 65
7.2 Future Work . . . 65
7.2.1 New Concept of Operations . . . 66
7.2.1.1 Mode 2 . . . 67
7.2.1.2 Mode 3 . . . 67
7.2.2 Alternate Operations . . . 68
References . . . 69
Appendices . . . 71

List of Figures

Page

2.1 Hysteresis loop for an ideal hard magnetic material 6
2.2 Possible dipole moments for a torquer pair 6
2.3 Quantization function for LPMT 7
2.4 Bias windows for alignment mode 8
2.5 Bias window offset 9
2.6 Bias regions for mode 2 9
2.7 Bias region for mode 3 10
2.8 Possible dipole moments for two torquer pairs 11
2.9 Quantization function for flight LPMTs 11

3.1 The ARC mechanical setup 15
3.2 Back and front sides of the Solar Panel Board showing the magnetometer location. . 16
3.3 The Alaska Research CubeSat (ARC) Attitude Control and Determination System (ACDS) board. 17
3.4 Block Diagram of the CubeSat ACDS system 19
3.5 Torquer locations within the ARC 20
3.6 The ARC ACDS torquer locations. 21
3.7 Torquer driver schematic 23
3.8 Torquer driving waveform showing multiple torquer flips. 24
3.9 Setup for measurements shown in figures 3.8 and 3.10. The Hall effect sensor on the end of the torquer is shown in detail. 25
3.10 Torquer saturation 26
3.11 Torquer feedback comparators 27
3.12 Magnetometer showing H_s and H_c 29

4.1 The Helmholtz cage used for ACDS testing 33
4.2 The Helmholtz cage computer and power supplies 34
4.3 Helmholtz cage block diagram 35
4.4 Helmholtz cage test field 36
4.5 Helmholtz cage test error 37

5.1 ACDS software overview 39
5.2 Overall Software Block Diagram 40

Page

6.1 Magnetic field offsets due to changing torquer states as measured by different Solar Panel Boards (SPBs) 46
6.2 Box plot showing torquer offset variations for the X-axis of the Y+ SPB 49
6.3 Box plot showing torquer offset variations for the Z-axis of the Y+ SPB 50
6.4 Y+ torquer compensation test showing that the correction provides a large amount of improvement over the uncorrected values 53
6.5 Graph showing Y+ torquer compensation error plot 54
6.6 Test of the calibration as performed by the ACDS board 56
6.7 Error plot for figure 6.6 showing torquer states 58
6.8 Plot of magnetic field errors for each axis 59
6.9 Simulation of torquer output with spinning magnetic field 61
6.10 Test of the B-dot controller 63

7.1 Overall Software Block Diagram 66
7.2 Mode 2 Torque Algorithm Block Diagram 67
7.3 Mode 3 Torque Algorithm Block Diagram 68

List of Tables

Page

5.1 ACDS operations data format 43
5.2 ACDS Beacon Data format 44

6.1 Information from `offsetplot`. All numbers are in Gauss 47
6.2 RMS errors in Gauss for ACDS system calibration test 57

List of Appendices

Page

Appendix A Offset Plots 71
Appendix B Calibration Equation Derivation 81

List of Acronyms

		Page
ADC	Analog-to-Digital Converter	27
ACDS	Attitude Control and Determination System	1
AMR	Anisotropic MagnetoResistance	27
ARC	Alaska Research CubeSat	2
ASGP	Alaska Space Grant Program	2
BCR	Battery Charge Regulator	16
CDS	CubeSat Design Specification	1
CDH	Command and Data Handling	39
COMM	Communication System	15
CRC	Cyclic Redundancy Check	55
DAQ	Data AQuisition	34
EPS	Electrical Power System	15
GPIB	General Purpose Interface Bus	33
GPIO	General Purpose Input/Output	
LEDL	Launch Environment Data Logger	15
NSF	National Science Foundation	1
I2C	Inter-Integrated Circuit	18
IMG	Imaging System	15
LPMT	Low Power Magnetic Torquer	1
MEMS	Microelectromechanical systems	18
MOSFET	Metal Oxide Semiconductor Field Effect Transistor	22
PCB	Printed Circuit Board	2
P-POD	Poly-Picosatellite Orbital Deployer	7
RAM	Random Access Memory	31
SD card	Secure Digital card	17
SMO	Science Mission Objective	16
SPB	Solar Panel Board	15

Page

SPI Serial Peripheral Interface 18

SSEP Space Systems Engineering Program 2

USB Universal Serial Bus 15

Acknowledgments

The work in this thesis would not be possible without help and support from others. The CubeSat renderings for this thesis were primarily done by Patrick Wade with help from the rest of the ARC mechanical team. The Solar Panel Boards were designed by Morgan Jonsen and Chic O'Dell. Most of the boards on the test setup were populated by Chic O'Dell. I would like to thank Dr. Hawkins and Dr. Thorsen for their support of projects that I have worked on over the years. The work in this thesis was supported by the Alaska Space Grant Program.

Chapter 1
Introduction

This thesis describes a CubeSat Attitude Control and Determination System (ACDS). The system uses Low Power Magnetic Torquers (LPMTs), which are torquers with a hard magnetic core that can be flipped by pulsing current through a coil that wraps around the core. This allows the CubeSat attitude to be controlled using less power than would be consumed using proportional torquers.

1.1 CubeSats

Over the past few decades, electronic devices have continued to get smaller, cheaper and more powerful. This has made many things possible that were not feasible before. Recently, it has become possible for a large number of universities to build and launch their own small satellites.

The CubeSat Design Specification (CDS)[1] is a small satellite standard that many universities, private companies and NASA have used to help them put satellites into orbit cheaper and easier than before. The CDS defines, among other things, the size of the CubeSats, a 10-cm cube, and how they interface with the launcher. Because CubeSats are small and launched from a standard launcher, they are easy to integrate into the extra space left over on larger missions, making launches relatively cheap and easy to obtain.

In recent years, the number of CubeSat launches has grown significantly. CubeSats are launched by universities as well as government institutions like NASA and the Air Force and private companies. CubeSats were originally envisioned as an educational tool that would have the approximate capabilities of Sputnik. While CubeSats may have been conceived as primarily an educational tool, CubeSats have been the subject of National Science Foundation (NSF) grants [2].

1.2 CubeSat ACDS

Because CubeSats are small, they have limited surface area for power generation. Many CubeSats do not use deployable solar panels because of their complexity, volume and weight. The result is that CubeSats have to run on a limited power budget. Many CubeSat science missions require, at the very least, a stable attitude or a nadir pointing attitude. On larger satellites, reaction wheels and thrusters are often used for attitude control. For many CubeSats, the added mass, power and volume of reaction wheels and thrusters is not a viable option.

One common CubeSat attitude control method is to use the Earth's magnetic field. Magnetic attitude control systems are generally capable of less torque than thrusters and reaction wheels, but are smaller and lighter, making them a popular choice for many CubeSats. Because magnetic torquers produce different torque based on the angle between the torquer and the magnetic field,

a completely passive system can be built. This is a popular choice for systems that mainly need to be detumbled as it is small, light and simple.

1.2.1 Passive Control

In a passive system, a permanent magnet on the spacecraft forces an axis to align with the local magnetic field. Additional hysteresis material must be added to the spacecraft to dampen oscillations that occur because of a lack of friction in the space environment. The hysteresis material dissipates energy in a changing magnetic field, thereby reducing the angular velocity. The main drawback for a passive system is that there is no face of the satellite that is continually facing the Earth. Having a face that constantly points in the nadir direction is useful for communications and for Earth observation.

1.2.2 Active Control

Active magnetic control allows for different attitude options such as nadir pointing. Many CubeSats use air core magnetic torquers built into the solar panel Printed Circuit Board (PCB) such as in [3, 4]. The coils are made of spiral shaped traces in one or more layers of the PCB. To increase efficiency some CubeSats use wire wound torquers with a magnetic core. In both cases, however, torque is generated only when current is flowing through the coil.

1.2.3 Low Power Magnetic Torquer Control

The system described in this thesis falls somewhere in between passive and active systems. The LPMTs used in the design allows more control of the attitude than a passive system, yet uses less power than current active systems. The LPMT is a good fit for CubeSats because of its power savings.

1.3 Alaska Research CubeSat

Alaska Research CubeSat (ARC) is a CubeSat built by the UAF Space Systems Engineering Program (SSEP) and funded by Alaska Space Grant Program (ASGP). The main purpose of ARC is to give students a chance to work on real space flight hardware. The ARC has two mission objectives which pertain to the ACDS system. The first is to "Validate a novel low power ACDS"[5]. This will be accomplished by implementing a low power ACDS and collecting data on its performance. The second is to "Validate a high bandwidth communication system by obtaining images of changing snow/ice coverage in arctic regions"[5]. This will be accomplished by putting the satellite in a

nadir facing attitude so the camera takes pictures of the Earth and the communication antenna points toward the ground.

1.4 Attitude Control and Determination System

The basic ACDS algorithm and torquer specifications have been designed by Dr. Donald Mentch in his doctoral thesis[6]. The remaining work for this thesis is the design, implementation and testing of the hardware and software to realize the ACDS on the ARC. In addition some parts of the system have been changed from the way the system was originally proposed due to practical considerations.

Chapter 2 outlines the previous work on LPMTs and the alignment algorithm that will be used on the ARC.

Chapter 3 describes how the ACDS hardware is implemented on the ARC. This hardware was constructed based on the parameters outlined in [6], but the bulk of the design, implementation and testing, was done by the author of this thesis.

Chapter 4 describes the Helmholtz cage used for testing of the system. The Helmholtz cage is capable of generating a uniform magnetic field in any direction and is used for testing of the magnetometer.

Chapter 5 describes the software for the ACDS. The algorithm for ACDS was written by Donald Mentch for his doctoral work [6]. For this thesis, the algorithm was converted from a MATLAB simulation into C code in order to run on the ARC ACDS embedded system. Because of the change in platform, many support functions had to be written.

Chapter 6 describes the tests used to validate the performance of the ACDS hardware and software. Some of the basic ideas of verification were outlined in [6] as well as in conversations with Donald Mentch. Most of the testing and verification has been performed by the author of this thesis and adjusted accordingly to the revised implementation.

Chapter 2
Background

The LPMT was originally designed in 1971 by Dr. Michael Polites[7]. It was refined into a stabilization system by Dr. Donald Mentch for his masters degree in 2003 and further refined into a system for the ARC as his doctoral work in 2011[6]. The system described in [6] proposes a concept for 3-axis stabilization that is valid for a nearly circular and nearly polar Earth-orbiting satellite.

2.1 Low Power Magnetic Torquers

The ARC ACDS uses Low Power Magnetic Torquers (LPMTs) to create the torque needed to properly orient the CubeSat. The torque equation for all magnetic torquers, including the LPMTs, is shown in equation (2.1) where $\vec{m}$ is the magnetic dipole moment of the torquer and $\vec{B}$ is the local magnetic field. The torque produced is always perpendicular to the magnetic field and the dipole moment. Also the component of $\vec{m}$ that is perpendicular to $\vec{B}$ is the only component that produces torque,

$$\vec{\tau} = \vec{m} \times \vec{B}\,. \tag{2.1}$$

Other CubeSats use magnetic torque rods and coils which are governed by the same equation. The unique part about LPMTs is that the magnetic dipole moment comes from the core of the torquer and does not depend on a continuous current flow to create torque.

2.1.1 Hard Magnetic Core

The cores of the LPMTs are made of a hard magnetic material that ideally have a saturation curve that looks like figure 2.1. The hysteresis curve shows the dipole moment of the torquer, M, with respect to the driving force, H, which is proportional to the current through the solenoid. The response of the torquer depends on the driving force as well as the previous state of the torquer so that when there is no driving force the torquer response has two possible states depending on how the torquer was last driven. The LPMT operates by switching the cores between these saturation points, M_s and $-M_s$. The LPMT cores are driven into saturation using a capacitor to provide a current pulse through a solenoid surrounding the magnetic core. Even after current stops flowing through the solenoid the core will have a residual dipole moment of $\pm M_r$ depending on which direction the current was run in [6].

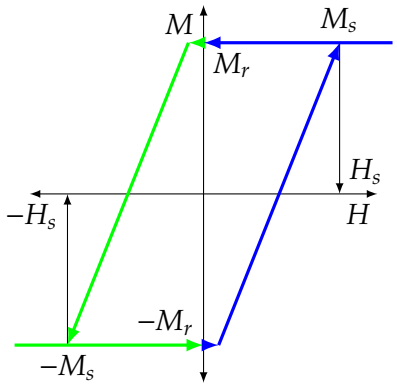


Figure 2.1. Hysteresis loop for an ideal hard magnetic material

2.1.2 Torquer Pair

The cores of the LPMTs are driven to saturation in the direction of the rod. This gives two possible states for each rod. Because in both states the rod generates a net dipole moment, the rods are operated in pairs so that three possible dipole moments can be produced as shown in figure 2.2, where M is the dipole moment of a single torquer.

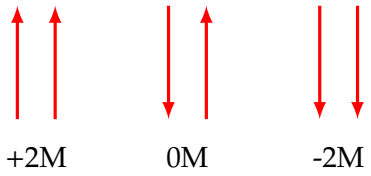


Figure 2.2. Possible dipole moments for a torquer pair

Because each LPMT pair can only produce three distinct values of dipole moment, the output of a LPMT algorithm must be quantized. Figure 2.3 shows the quantization function for two torquers, where $M_{command}$ is the commanded dipole moment from the algorithm and $M_{command_q}$ is the quantized dipole moment that takes on one of the values that the torquers can produce.

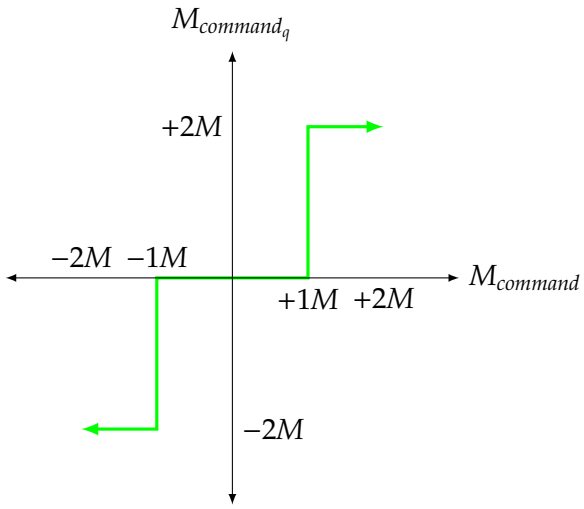


Figure 2.3. Quantization function for LPMT

2.1.3 Torquer Sets

For the initial design in [6] it was proposed to use a 10 second torquer flipping interval and two pairs of different torquers in each axis. One pair, called the primary torquers, was made of Alnico1 and had a dipole moment of 0.022 $\text{A}\cdot\text{m}^2$. This pair was to be used for the detumble phase where a larger dipole moment allows for more torque to slow the satellite faster. For the alignment phase the second set of torquers, the vernier torquers, were proposed. The vernier torquers were to have a dipole moment of 0.00011 $\text{A}\cdot\text{m}^2$.

2.2 Detumble Algorithm (Mode 1)

The purpose of the detumble algorithm is to reduce the initial tipoff rates. When CubeSats are ejected from the Poly-Picosatellite Orbital Deployer (P-POD) significant rotation rates can be induced.

To detumble ARC the control law shown in equation (2.2) is used. This is used by many CubeSats with conventional torquers to detumble. The significant difference in this case is that the magnetic dipole moment will be quantized due to the LPMTs,

$$\vec{m}_{command} = k\frac{\vec{\omega}_{error} \times \vec{B}}{\vec{B} \cdot \vec{B}}. \tag{2.2}$$

The original simulation transitioned to detumble once the rotation rate error was reduced below 0.002 $\frac{\text{rad}}{\text{sec}}$, corresponding to about 1.7 revolutions/orbit, from the rotation rates imparted on the CubeSat as it exits the P-POD. It was later suggested that the transition could run for a pre-determined time that was more than sufficient to detumble the satellite.

2.3 Alignment Algorithm

The job of the alignment algorithm is to align the ARC in the proper attitude and maintain this attitude for the rest of the mission. This is achieved using bias windows as shown in figure 2.4. In the equatorial windows the magnetic field is nearly parallel to the surface of the Earth and in the polar window the magnetic field is nearly perpendicular to the surface of the Earth. By biasing the torquers in such a way as to line up the appropriate axis in each window, three-axis alignment can be achieved. There is no window at the south pole because the magnetic south pole is located far from the geographical south pole making it necessary to know the direction of approach to determine the window location.

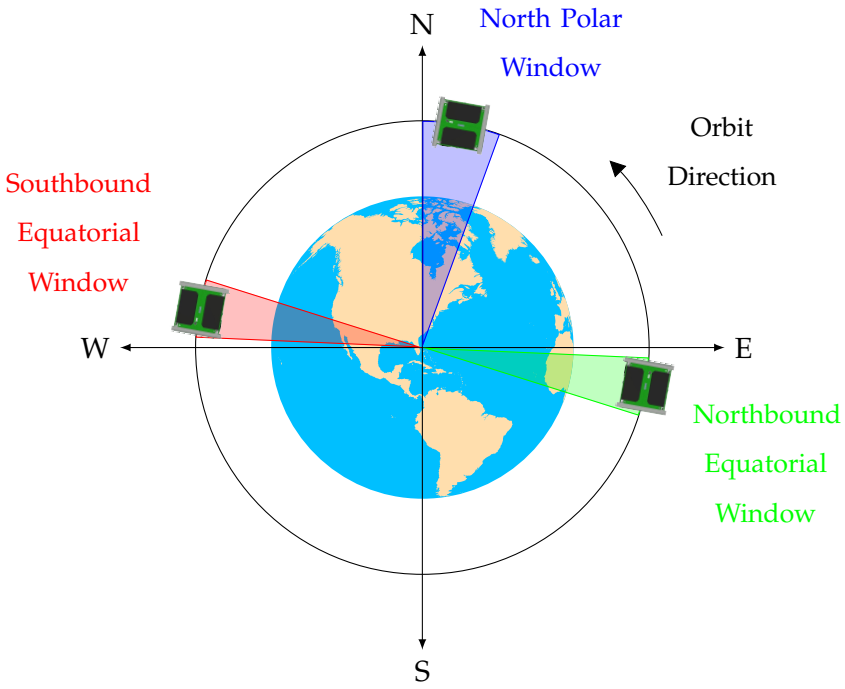


Figure 2.4. Bias windows for alignment mode

Figure 2.4 shows that the bias windows are offset so that they occur just before the equator or pole. The reason for this is shown in figure 2.5. If the torquers are biased before the satellite passes over the vertical (or horizontal) field lines then the tendency is to rotate in the desired direction. If the torquers are biased afterwards then the tendency is to rotate the satellite opposite the desired direction[6].

Figure 2.5 is similar to Figure 29 in [6]. The major change to note is that the axis definitions have been changed to match those of the ARC.

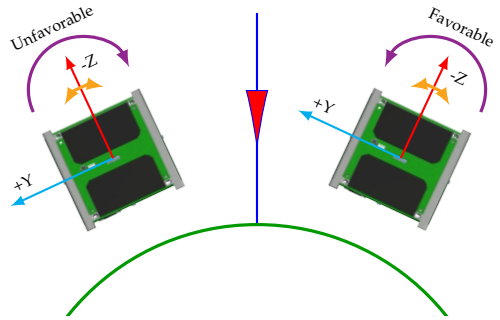


Figure 2.5. Bias window offset

2.3.1 Mode 2

Mode 2 is initiated at the end of the detumble phase. In mode 2 the torquers are biased as shown in figure 2.6 with the Y-axis torquers biased in the equatorial windows and the Z-axis torquers biased in the polar window. While the bias is active the algorithm in equation (2.2) is used to dampen out the oscillations caused by the bias. Outside the three bias windows all torquers are set to cancel each other and the ARC coasts.

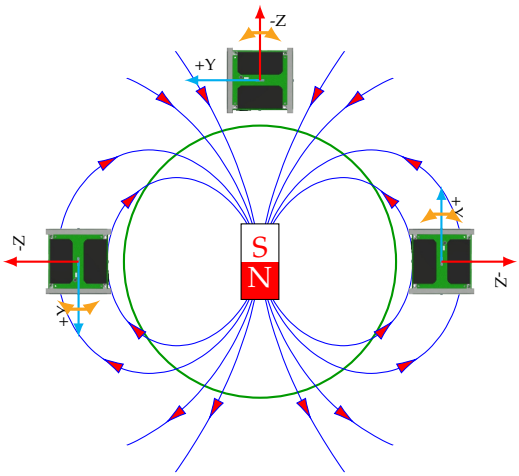


Figure 2.6. Bias regions for mode 2

The purpose of mode 2 is to get the satellite, which is in a random orientation, into the desired orientation. The original simulation transitioned from mode 2 to mode 3 after 10 orbits. This was shown to get the satellite into the desired orientation.

2.3.2 Mode 3

Mode 3 is initiated after mode 2 is complete. In mode 3 only the north pole bias window is used as shown in figure 2.7. Outside the bias window however the algorithm in equation (2.2) is used to maintain desired rotation rates. This is done to reduce power consumption during operation.

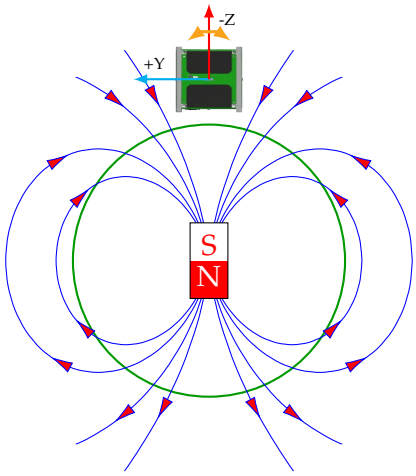


Figure 2.7. Bias region for mode 3

2.4 Simulation Results

The simulations in [6] showed that tipoff rates of 5°/sec could be arrested within two 98° inclination orbits. In mode 3 the simulations showed that the desired orientation was achieved within 2° in all axes using Alnico1 torquers for detumble and vernier torquers for alignment.

2.5 Concept of Operations for the Alaska Research CubeSat

The initial concept was proposed by Mentch in [6]. The concept was largely based on what was possible in simulation. The simulation allowed the satellite perfect knowledge of the rotation rates which are not trivial to calculate in a real system from measured magnetic fields. The rotation rates can be calculated from the magnetic field using a Kalman filter as suggested in [8] but this is beyond the scope of this thesis.

Difficulties were also encountered with the torquers. The vernier torquers were proposed to be $^1/_{200}$ the size of the primary torquers. This presented two problems: manufacturing and balancing. Finding a suitable, weaker, magnetic material for the torquers proved difficult as Mentch said "There isn't much of a market for crappy magnets." The solution that was proposed was to coat a non-magnetic rod with a very thin layer of magnetic material using a vacuum deposition process.

There were difficulties fabricating torquers and later tests showed that there is enough variation in the dipole moment of the primary torquers to overwhelm the vernier torquers.

Because of these difficulties the ACDS was re-scoped to focus on detumble. Instead of one pair of primary torquers and one pair of vernier torquers, two pairs of primary torquers are used. This gives five possible states as shown in figure 2.8.

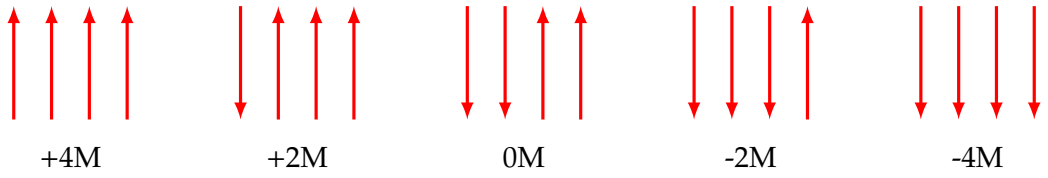


Figure 2.8. Possible dipole moments for two torquer pairs

The quantization function shown in figure 2.3 has been changed to the one in figure 2.9 because the number of torquers has increased. The increased number of torquers increases the maximum torque but doesn't change the minimum possible torque.

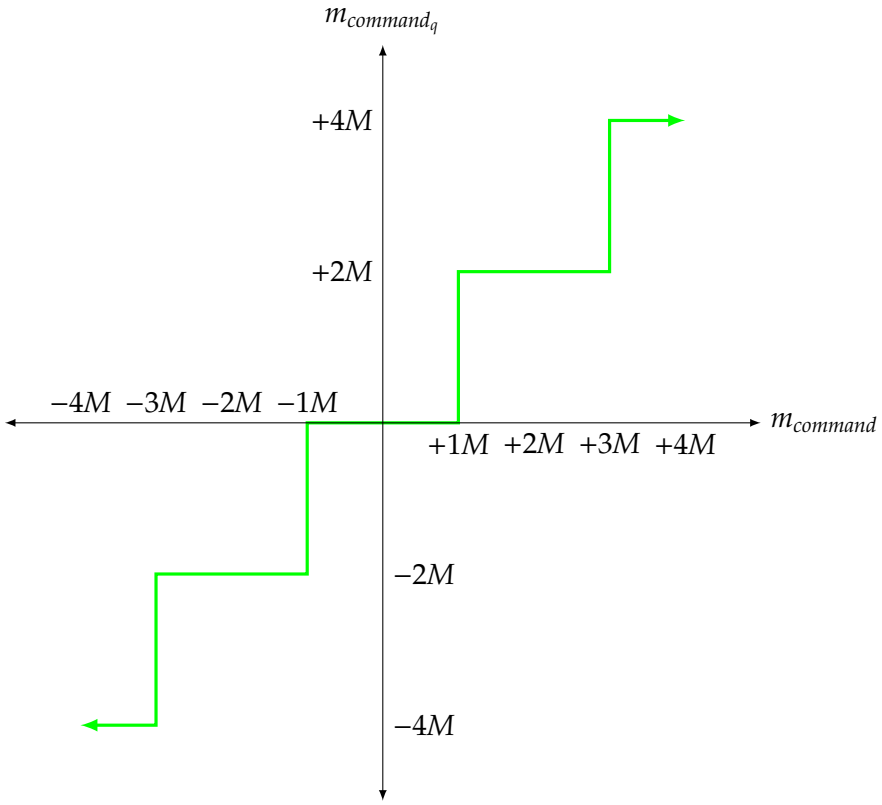


Figure 2.9. Quantization function for flight LPMTs

Adding a second set of primary torquers allows a larger dipole moment during the detumble

phase but does not help during the alignment phase when smaller torques need to be generated. The original design used a 10 second charge time which meant that once a torquer was flipped it would be torquing the satellite for a minimum 10 seconds. To help mitigate the use of more powerful torquers the charge time was changed to 1 second. This means that the minimum time that the torquers can be torquing is reduced, reducing the overall change in angular velocity of the satellite. The reduction in charge time is only by a factor of 10 but the minimum torque was increased by a factor of 200 so the pointing accuracy of the satellite will suffer.

Rotation rate determination also proved a challenge for the on-orbit implementation. The satellite in simulation had full knowledge of the rotation rates. In [6] it was proposed that rotation rates could be directly calculated using the magnetic field.

Using the magnetic field to determine rotation rates does not give the full rotation rate vector. Instead the rotation rate vector is in effect projected onto the magnetic field, eliminating any rotation rate around the magnetic field. Originally it was believed that this would not be a problem because the torquers can't be used to stop rotation around the magnetic field. Simulations showed, however, that when the rotation rate vector was projected onto the magnetic field then the attitude control did not work.

The other problem with rate determination is that the desired rotation of the satellite is to rotate once every orbit in one axis and no rotation in the other axes. The expected orbital period is about 90 minutes so this gives a rotation rate of 0.011 rpm or 1.16 mrad/sec. In 10 seconds a satellite rotating at this rate will rotate about 20 μ°. For a satellite rotating in a 300 mGauss magnetic field this results in a 60 μGauss field change every 10 seconds. Thus in order to measure the rotation rates of the satellite the magnetic field of the Earth must be measured with an accuracy that is on the order of 10's of μGauss.

In [8] a Kalman filter is used to calculate rotation rates from magnetic field measurements. This solution solves both problems. By taking into account the spacecraft dynamics, the full rotation rate vector can be estimated. Because the rates come from the estimated system state they are less noisy than those from a single set of magnetic field measurements. The problem is that Kalman filters are not trivial to get working and must correctly model the system to function correctly. This is beyond the scope of this thesis so the alignment phase of the attitude control system is not implemented.

2.5.1 B-dot Control

The algorithm in equation (2.2) can't be used for flight because the rotation rates, $\vec{\omega}$, are not known. The B-dot algorithm uses the magnetic field time derivative, $\dot{\vec{B}}$, as its only input so full knowledge

of spacecraft rotation rates is not necessary for detumble,

$$\vec{m}_{command} = C\dot{\vec{B}}. \tag{2.3}$$

Equation (2.3) shows the B-dot control law. Because $\dot{\vec{B}}$ results largely from the spacecraft rotation, it is largely perpendicular to $\vec{B}$. This results in a dipole moment that is perpendicular to the magnetic field which is desirable for efficiency reasons. The gain C is a scalar value less than zero that is used to tune the algorithm. Large values of C increase the sensitivity of the algorithm and allow it to detumble to a lower rotation rate. Lower values of C result in less sensitivity to noise. [9]

Chapter 3

Hardware Implementation of the Attitude Control and Determination System (ACDS)

As stated in chapter 2 the system design from [6] has been modified for the implementation in this thesis. The only hardware change from [6] is to use four Alnico1 torquers in each axis instead of two Alnico1 and two vernier torquers as proposed. This does not effect the electrical hardware as both torquer types are the same size and are flipped by the same circuit.

The ACDS board for the ARC only contains the processor, torquers and drivers. The sensors are located off board and managed by the Launch Environment Data Logger (LEDL). The sensor data is sent over the bus to the ACDS processor.

3.1 Mechanical Configuration of the Alaska Research CubeSat (ARC)

Figure 3.1 shows the mechanical design of the ARC. The design has a central board stack with solar panels and rails attached to rings that are on the top and bottom of the board stack. The boards in the stack are connected by a 104-pin "stack through" header that is located on the -Y edge of each board. The header distributes power and allows for communication between boards.

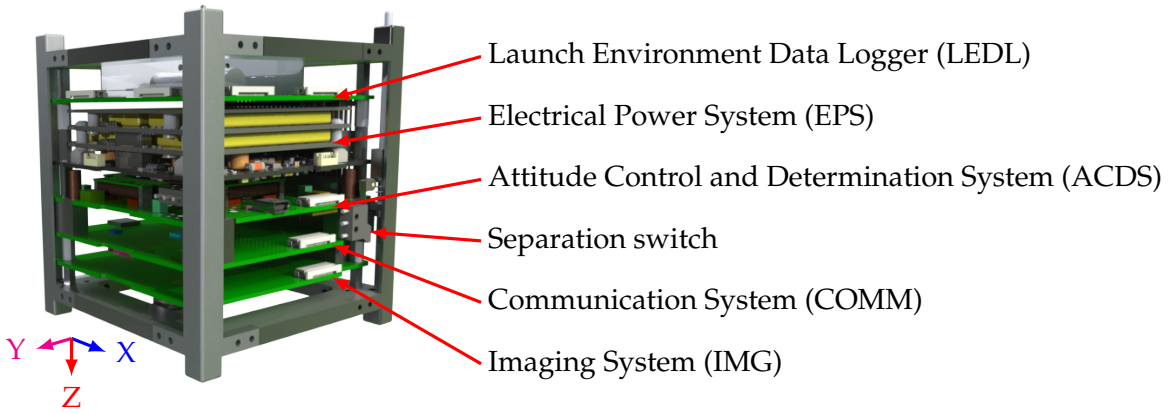


Figure 3.1. The ARC mechanical setup

The ACDS board is located 3rd from the bottom of the stack. This puts it in the center of the board stack. Because the solar cells cover all of the side faces except a small strip in the center, this makes the ACDS board the only place to put the pull pin and Universal Serial Bus (USB) connector that need to be accessed while ARC is in the P-POD. The separation switch connections are also located on the ACDS board, because the separation switch is attached to one of the torquer standoffs.

For clarity the Solar Panel Boards (SPBs) are not shown in figure 3.1. The SPBs are attached to mounting holes in the rails and are the outer faces of the ARC. In addition to the solar cells the

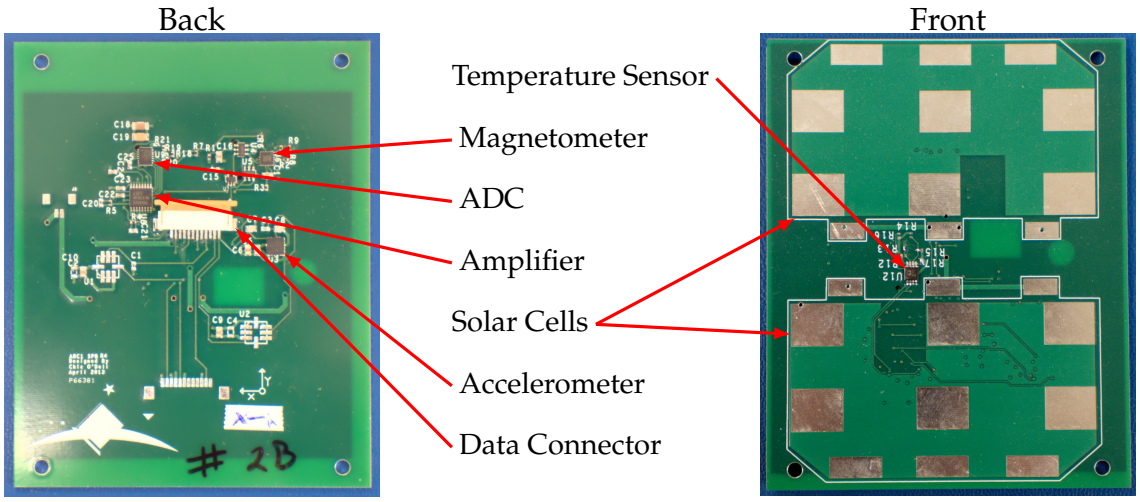


Figure 3.2. Back and front sides of the Solar Panel Board showing the magnetometer location.

SPBs also contain sensors that are used by the LEDL and ACDS. Figure 3.2 shows the back side of a side SPB with the sensors labeled, and the front side showing the temperature sensor and the pads where the solar cells will mount. The top and bottom SPBs are similar to the side SPBs except they lack accelerometers and include parts for the antennas and antenna deployment.

The core board stack contains the major subsystems for the ARC. Some of these subsystems are designed to carry out a specific Science Mission Objective (SMO) while others provide a general support role. The subsystems are as follows:

LEDL : One of the SMOs on ARC is to "Characterize the thermal and vibration environment inside the launch vehicle from ignition to orbit insertion"[5]. The LEDL does this by using a separate battery to operate during the launch phase and log data from accelerometers and temperature sensors.

EPS : The EPS manages the electrical power on ARC. The solar cells connect to Battery Charge Regulators (BCRs) which have peak power point tracking and output the proper voltage to charge the batteries. The voltage from the batteries is regulated down to 5 V and 3.3 V. The EPS supplies the unregulated battery voltage along with the 5 V and 3.3 V rails to the rest of the satellite. The EPS is the only system on the ARC not built at UAF. The EPS is designed and built by Clyde Space[10].

ACDS : One of the SMOs on ARC is to "Validate a novel low power Attitude Control and Determination System (ACDS)"[5]. This is the system discussed in this thesis.

Separation Switch : This switch is screwed into one of the torquer standoffs and cuts off power

to the CubeSat when it is in the P-POD.

COMM : The COMM allows for two-way communication with the ground. A 9600 bps command/beacon up/down link and a 38400 bps data down link are used. The 9600 bps link operates on the 436 MHz band and uses a tape measure monopole. The 38400 bps link operates on the 2.4 GHz and uses a ceramic patch antenna mounted on the bottom of the ARC.

IMG : One of the SMOs on the ARC is to "Validate a high bandwidth communication system by obtaining images of changing snow/ice coverage in arctic regions"[5]. The imager has a camera and an Secure Digital card (SD card). The camera is used to take pictures and store them on the SD card.

The EPS is designed and manufactured by Clyde Space as a generic CubeSat EPS. The EPS is compatible with CubeSat Kit[11] hardware which is one of the main standards for off-the-shelf CubeSat hardware. Because of this the rest of the ARC subsystems must be compatible with the CubeSat Kit hardware. All boards have the same header and mounting hole locations and similar outlines so that they all fit together in the CubeSat.

3.1.1 Alaska Research CubeSat Attitude Control and Determination System Hardware

The ACDS board has the same overall outline and hole pattern as the rest of the ARC subsystems but it has 4 notches cut out to accommodate the Z-axis torquers and torquer standoffs. Additionally, there is a bump-out for the USB connector so that it can be closer to the access port to allow a USB cable to be easily plugged in.

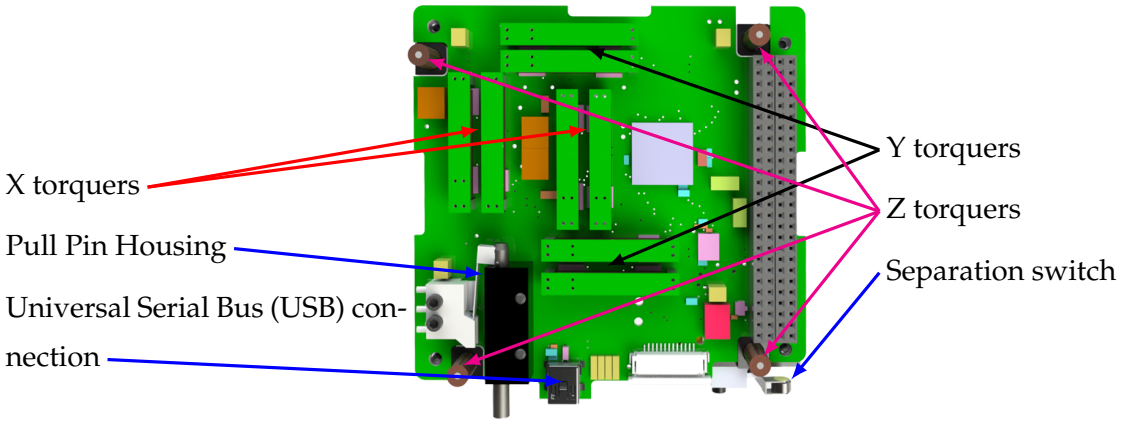


Figure 3.3. The ARC ACDS board.

In addition to the ACDS hardware, the ACDS board also is the connection point for the pull pin, separation switch, and USB connection port. These are located on the ACDS board because it is the only board that is accessible through the SPB. The pull pin, separation switch, and USB connection connect directly to the header and are not connected to the core ACDS hardware. The USB Port is used for charging and satellite diagnostics during testing and on the ground. The Pull Pin, when inserted, keeps the satellite powered off until it is inserted into the P-POD after which the separation switch keeps the satellite powered off until orbital insertion.

The Z-axis torquers are housed inside the standoffs which are located on the four corners of the board. The ACDS board shape has been modified from the standard board shape to accommodate the torquers and standoffs. This is visible in figure 3.3. The notches are designed to fit the z-axis torquer standoffs and keep them aligned in the proper orientation. The torquers fit into a hole drilled into the standoffs and are held in place with epoxy.

3.1.2 Bus Communication

The subsystems of the ARC will communicate with each other using the ARCBus. The ARCBus consists of shared connections between the subsystems to transmit commands and data. The ARCBus will be primarily used by the ACDS to get sensor data from the LEDL and send and receive data to the ground station through the COMM system. Commands are transmited using an Inter-Integrated Circuit (I2C) bus and large blocks of data are transfered using a Serial Peripheral Interface (SPI) bus which is negotiated using the I2C bus.

3.2 Components of the Attitude Control and Determination System

The block diagram for the CubeSat hardware used to implement the ACDS is shown in figure 3.4. The hardware necessary for the ACDS is spread over multiple subsystems of the ARC. The ACDS board contains the torquers and driving hardware along with the microcontroller which will run the attitude control algorithm. There is one two-axis magnetometer located on each of the six SPBs. These are used by the ACDS to calculate rotation rates and calculate the magnetic dipole moment required to generate the necessary torque. Spreading the magnetometers across all six faces should give some degree of noise immunity and redundancy. The LEDL board also contains Microelectromechanical systems (MEMS) angular rate sensors as a redundant measurement of the rotation rate of the ARC. The sensors on the SPBs are read by the LEDL, which then forwards the magnetometer and angular rate measurements to the ACDS.

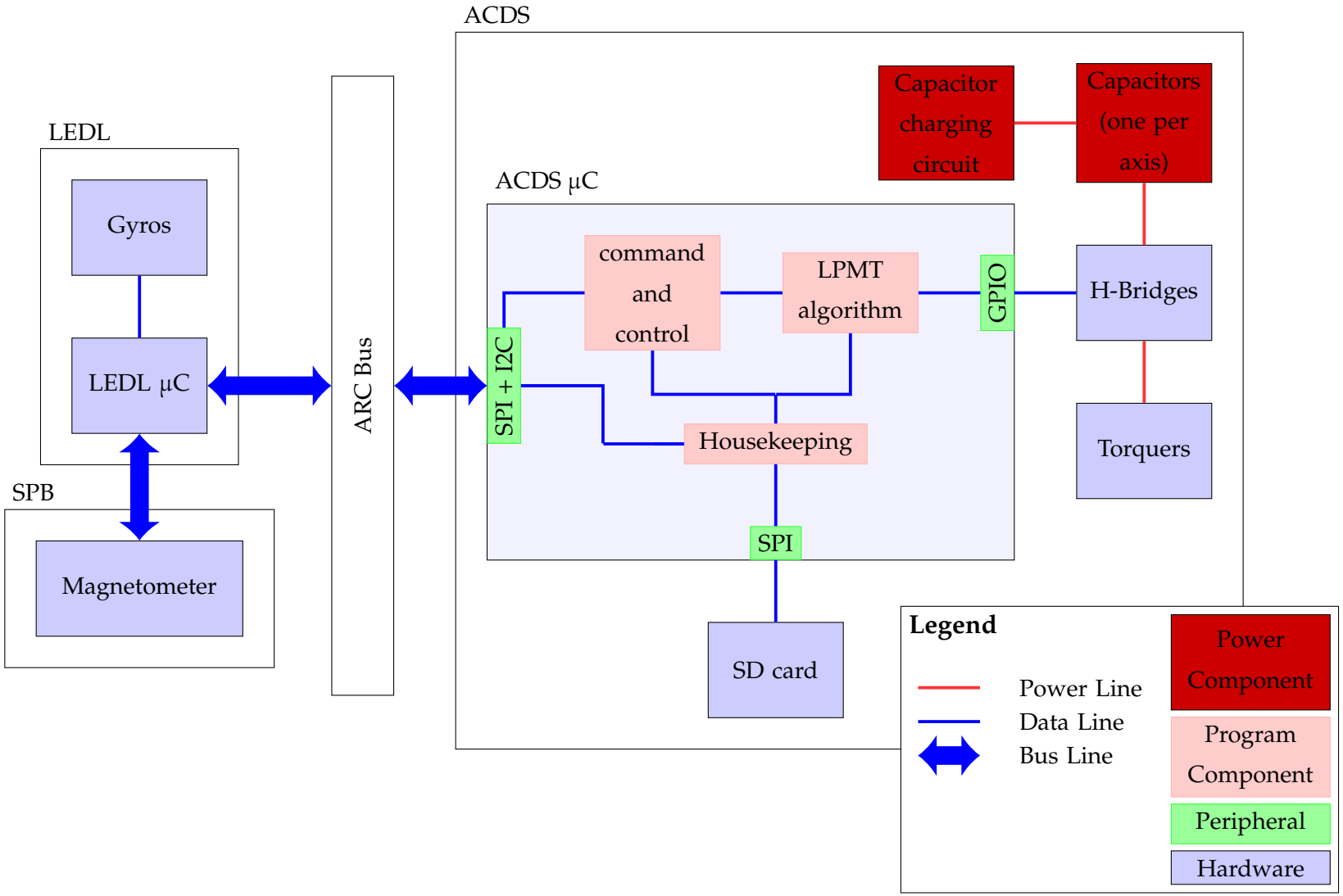


Figure 3.4. Block Diagram of the CubeSat ACDS system

3.3 Torquers

Figure 3.5 illustrates the torquer locations within the ARC. The torquers consist of a hard magnetic core surrounded with a coil of wire. There are a total of twelve torquers on the ARC, with four in each axis. The torquers are flipped using a driving circuit that causes a current pulse to flow through the coil. The driving hardware for the torquers is designed to flip one torquer in each axis every second.

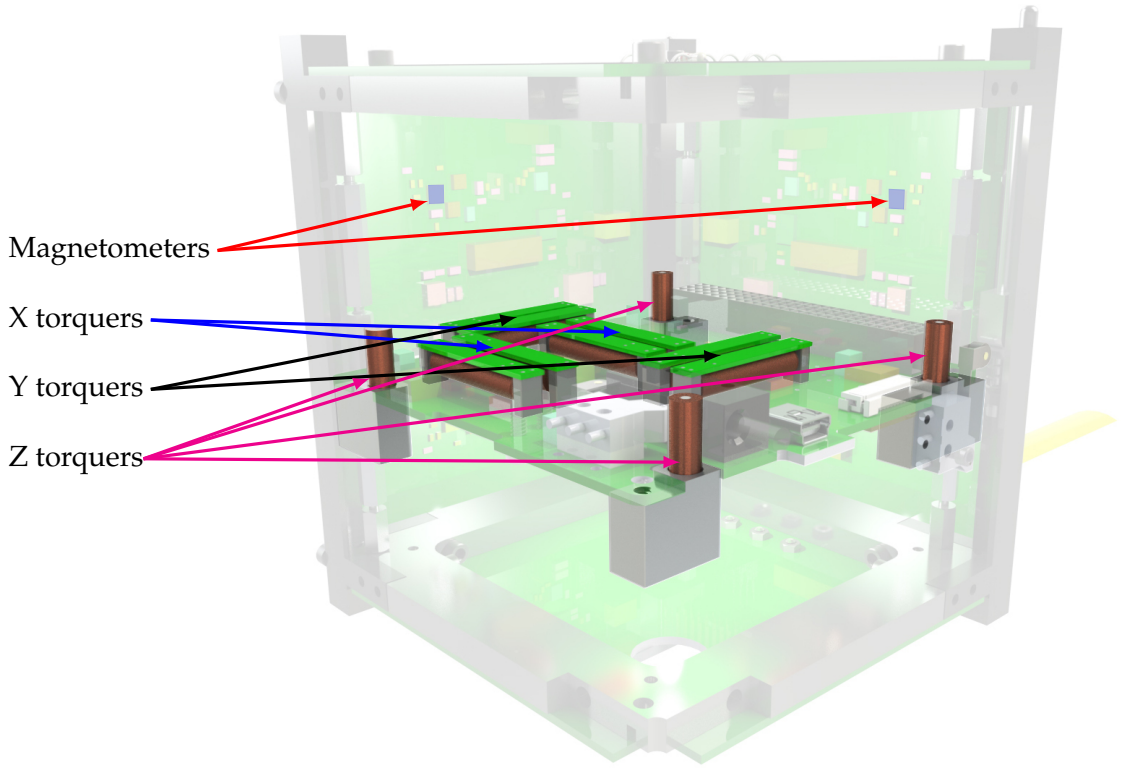


Figure 3.5. Torquer locations within the ARC

Figure 3.6 shows the location and identification designator of each torquer on the ACDS board. This is how the torquers are referred to in software.

3.3.1 Cores

Most magnetic torquers use either an air core or a soft magnetic core. Ideally both an air or a soft core only produce torque while a current is running through the coil. A soft core will produce more torque for the same geometry and current compared to an air core. However, soft magnetic materials have some hysteresis that will continue to produce undesired torque after current stops flowing. Hard magnetic materials are chosen to have a large amount of hysteresis so that torque is still produced after the current stops flowing in the coil.

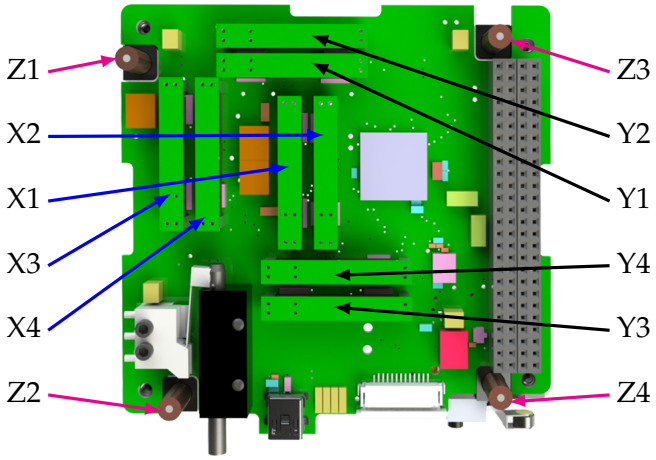


Figure 3.6. The ARC ACDS torquer locations.

The torquer cores for the ARC are made of Alnico1, which is a hard magnetic material. Core materials are chosen for their residual induction (B_r) and their coercive force (H_c). The torque produced for a given torquer geometry is proportional to B_r. This means that B_r dictates the range of torques that are feasible. If B_r is too small, then the torquers will have to be too big to fit in the satellite. If B_r is too large, then it may not be possible to manufacture torquers that are small enough to give the desired torque. The current required to flip the torquer for a given coil and core geometry is determined by H_c. It is desirable to have a low H_c so less energy is used for each flip. However, if H_c is too low then the torquer can be flipped inadvertently.

3.3.1.1 Core Sizing

The proposed torquer cores consisted of a large and a small pair in each axis. The large pair were to be made of Alnico1 and produce a magnetic dipole moment of $0.022\,\mathrm{A}\cdot\mathrm{m}^2$. The small pair was proposed to have an inert core with a thin permalloy coating to give it a dipole moment of $0.00011\,\mathrm{A}\cdot\mathrm{m}^2$. All torquer cores were to be the same size: one inch long $1/16$-inch diameter rod.

As discussed in [6] the large cores were sized by first determining the desired torque for each torquer, and then calculating the necessary dipole moment. Simulation was used to refine the required dipole moment. Alnico5 was the core material that had been used in previous experiments for much larger cores[6]. The minimum manufactured size of Alnico rods is $1/16$-inch. It is desirable to have a length-to-diameter ratio greater than 10, which fixes the minimum torque that can be achieved. The torque using Alnico5 was larger than the desired torque so Alnico1 was used instead, which has a lower residual induction.

The proposed size of the vernier torquers was $1/200$th the torque of the Alnico torquers. The

fabrication of the proposed vernier torquer was not completed and even if it had it is not clear that testing could have been done to quantify the dipole moment. In addition to fabrication and testing issues there is the problem of balancing the primary cores. Because of theses difficulties the original design for the small torquers was dropped and an extra set of Alnico torquers was added in their place. The charge time was also shortened from 10 seconds to 1 second.

3.3.2 Coils

The coils used to drive the torquers are made from 248 turns of 26 AWG wire in four layers. The current spike to flip the torquer peaks around 11 A which is significantly higher than 26 AWG can handle with continuous current. However, because the pulse duration is short it is not a problem.

A 1-inch long solenoid with 11 A through 248 turns of wire has a coercive force of 107 kA/m. This is significantly higher than the coercive force of Alnico1 which is 37 kA/m[12]. This shows that the solenoids on the ARC are more than capable of flipping a torquer.

3.3.3 Drivers

The schematic of the circuit used to drive a pair of torquers is shown in figure 3.7. Each pair of torquers is driven by three complimentary pairs of Metal Oxide Semiconductor Field Effect Transistors (MOSFETs). The MOSFETs are driven by three pairs of MOSFET drivers that are controlled by the ACDS microcontroller. The capacitor C1 is shared between both pairs of torquers in each axis giving a total of three capacitors on the ACDS board. This means that only one torquer in each axis can be flipped each second.

In the idle state, all of the MOSFETs are off and the torquer coils are floating. This prevents unwanted currents from flowing in the torquers. If MOSFETs are left on, providing a current path between the coil terminals, currents can be induced in the coils as the satellite turns and the magnetic field in the torquers changes. To flip a torquer, one end of the torquer is connected to ground using one of the N-Channel MOSFETs and the other end is connected to capacitor C1 using one of the P-Channel MOSFETs. After a 2 ms delay, the MOSFETs are switched off and C1 begins to recharge.

3.3.3.1 Driving Waveform

Figure 3.8 shows the driving waveform when driving a torquer multiple times in the same direction. The first torquer flip is shown in blue and the last flip is shown in red. The first waveform is the only waveform that differs significantly from the subsequent waveforms because the torquer is

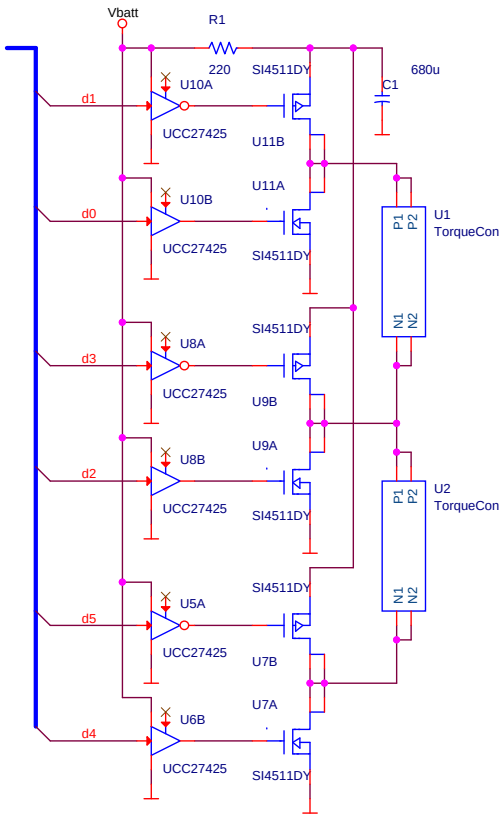


Figure 3.7. Torquer driver schematic

fully saturated in the first flip, thus demonstrating that the driving circuit is adequate to completely flip the torquers using a single 2 ms pulse.

The magnetic field waveform for the first flip starts at about +125 Gauss. This is because the torquer was flipped in the opposite direction before the test started. The subsequent waveforms start at about -125 Gauss. This is because the first flip drives the torquer into saturation and flips the torquers direction of magnetization. For all of the flips after the first flip, the change in magnetic field waveform is only due to the torquer coil and not to an actual torquer flip.

The current waveforms also show changes due to torquer saturation. For the first flip the current initially increases at a slower rate. This is because, on the Alnico1 B-H curve, the slope is steeper closer to the B-axis resulting in a higher inductance. As the torquer reaches saturation, the inductance of the torquer goes down and the rate of change of current increases. For the subsequent torquer flips, the torquer is already saturated so the inductance is low and the driving current reaches its peak much faster.

Figure 3.9 shows the hardware used to take the data for figures 3.8 and 3.10. The ACDS board

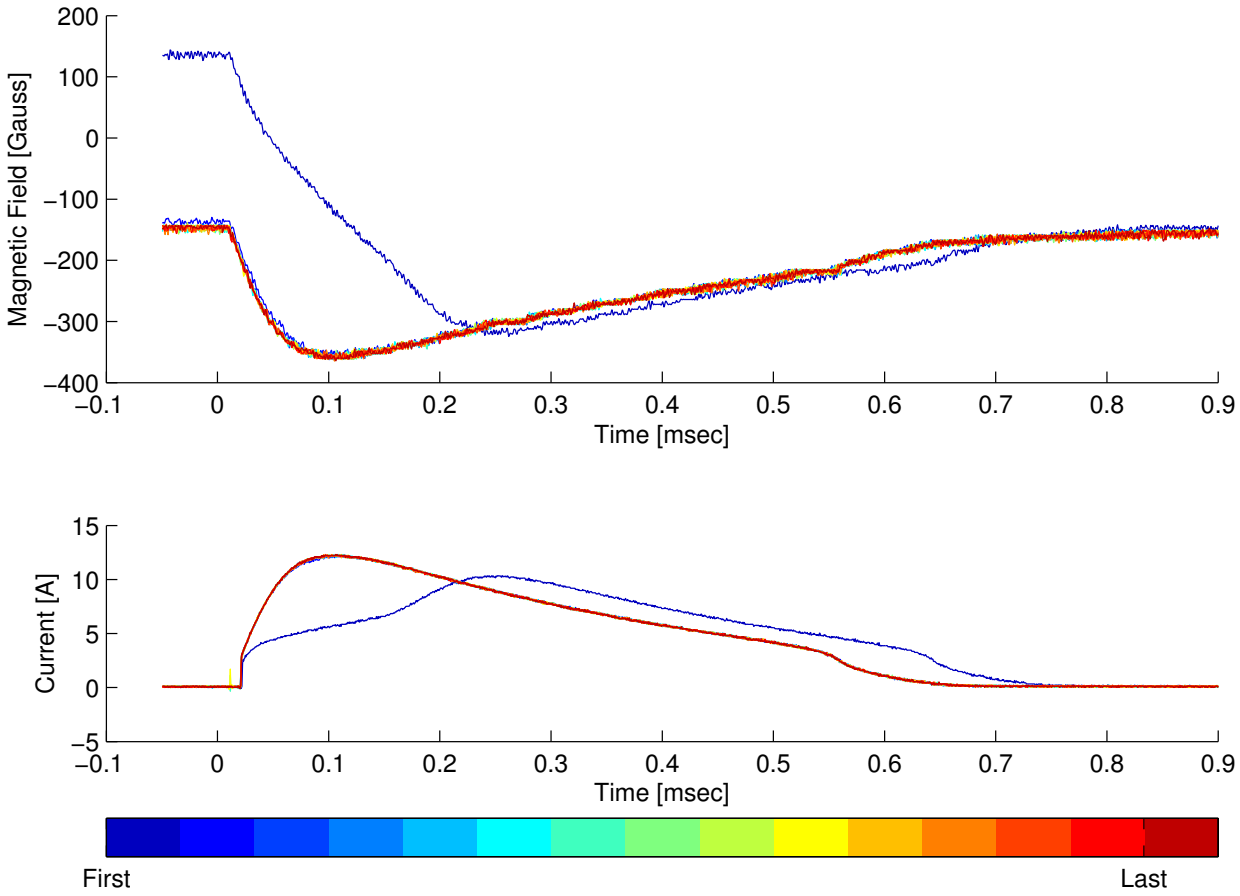


Figure 3.8. Torquer driving waveform showing multiple torquer flips.

was used to drive the torquers. To communicate with the ACDS, another microcontroller board was used along with a USB-to-serial converter. The torquer connects to the ACDS board through a current sensor which is read by the oscilloscope. The torquer is placed inside a plastic housing with an attached Hall effect sensor to read the magnetic field. The Hall effect sensor is located against the end of the torquer and is read by the oscilloscope. A pin on the ACDS microcontroller is used to trigger the scope which is automatically configured by a MATLAB script that communicates with the ACDS board and the oscilloscope.

The measured field that the Hall effect sensor reads is highly dependant on the distance from the torquer. The residual induction of Alnico1 is 7200 Gauss [12]. For the radius ($1/32$ inch) and length of the torquer (1 inch), the calculated magnetic field of the torquers is 155 Gauss at 0.05 in from the face of the torquer in the torquer axis[13]. This is close to what is shown in figure 3.8. As the distance from the face increases to 0.1, 0.2 and 0.3 inches the field decreases to 46, 12 and 5 Gauss, respectively.

Figure 3.10 shows how the torquer field changes as the number of flips increases. Both graphs

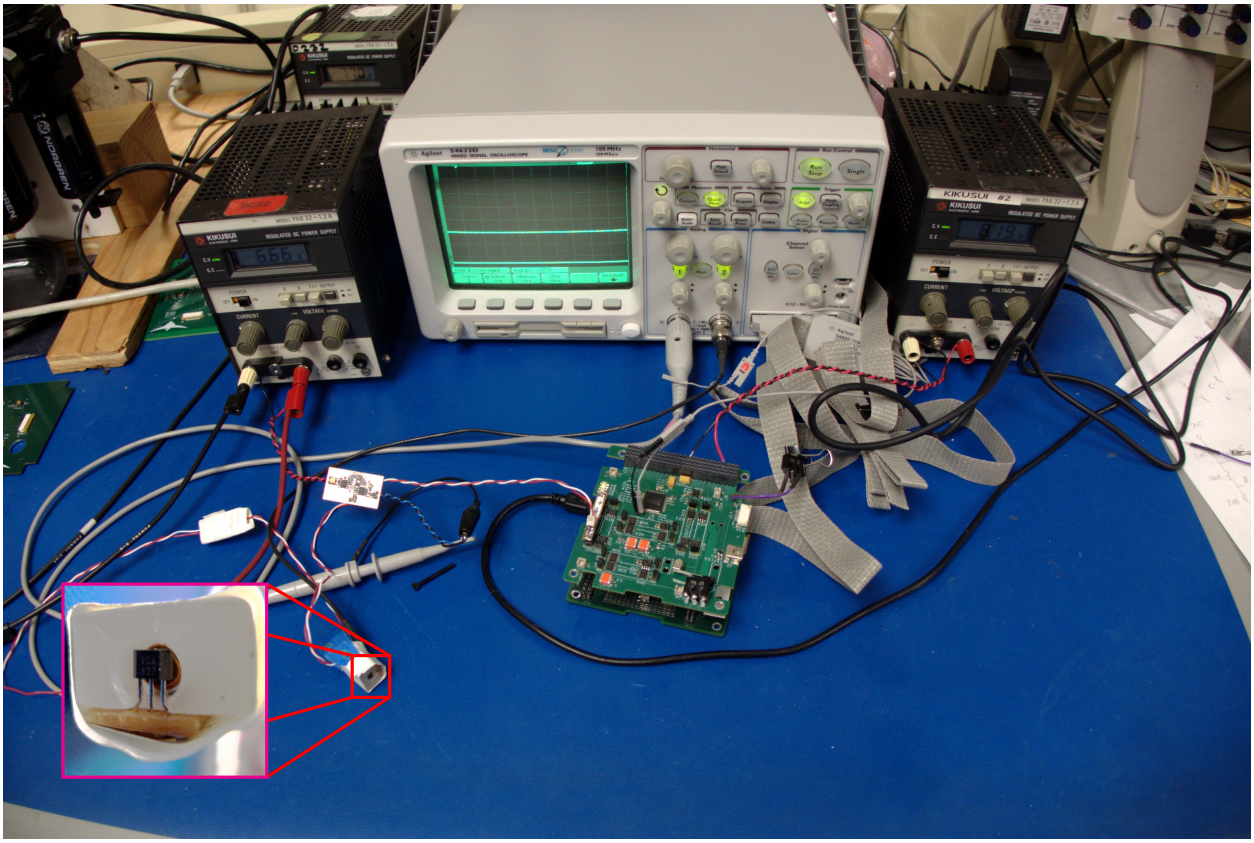


Figure 3.9. Setup for measurements shown in figures 3.8 and 3.10. The Hall effect sensor on the end of the torquer is shown in detail.

show the same data, with the lower graph zoomed in to better show the points after the first flip. The 0% point is found by averaging 20 samples from the beginning of the first magnetic field waveform in figure 3.8. The rest of the points are found by averaging the last 20 samples of the magnetic field waveforms from figure 3.8. The first flip gets the torquer field to within 97% of the final value and after two flips it has reached the final value.

3.3.4 Torquer Diagnostics

Figure 3.11 shows the torquer feedback comparators that are used to detect if the torquers and drivers are functioning properly in flight. Vcap is the voltage from capacitor C1 in figure 3.7. The comparator outputs, FB0 and FB1, have open drain outputs that are pulled up to logic hi by internal resistors on the MSP430 inputs. U12A is used to determine if the capacitor is charged and U12B is used to detect if the capacitor is discharged. The comparators are checked both before and after a torquer flip to see if the capacitor was charged before an attempted flip and discharged after. If the capacitor did not discharge, then the torquer did not flip, probably because of a bad connection

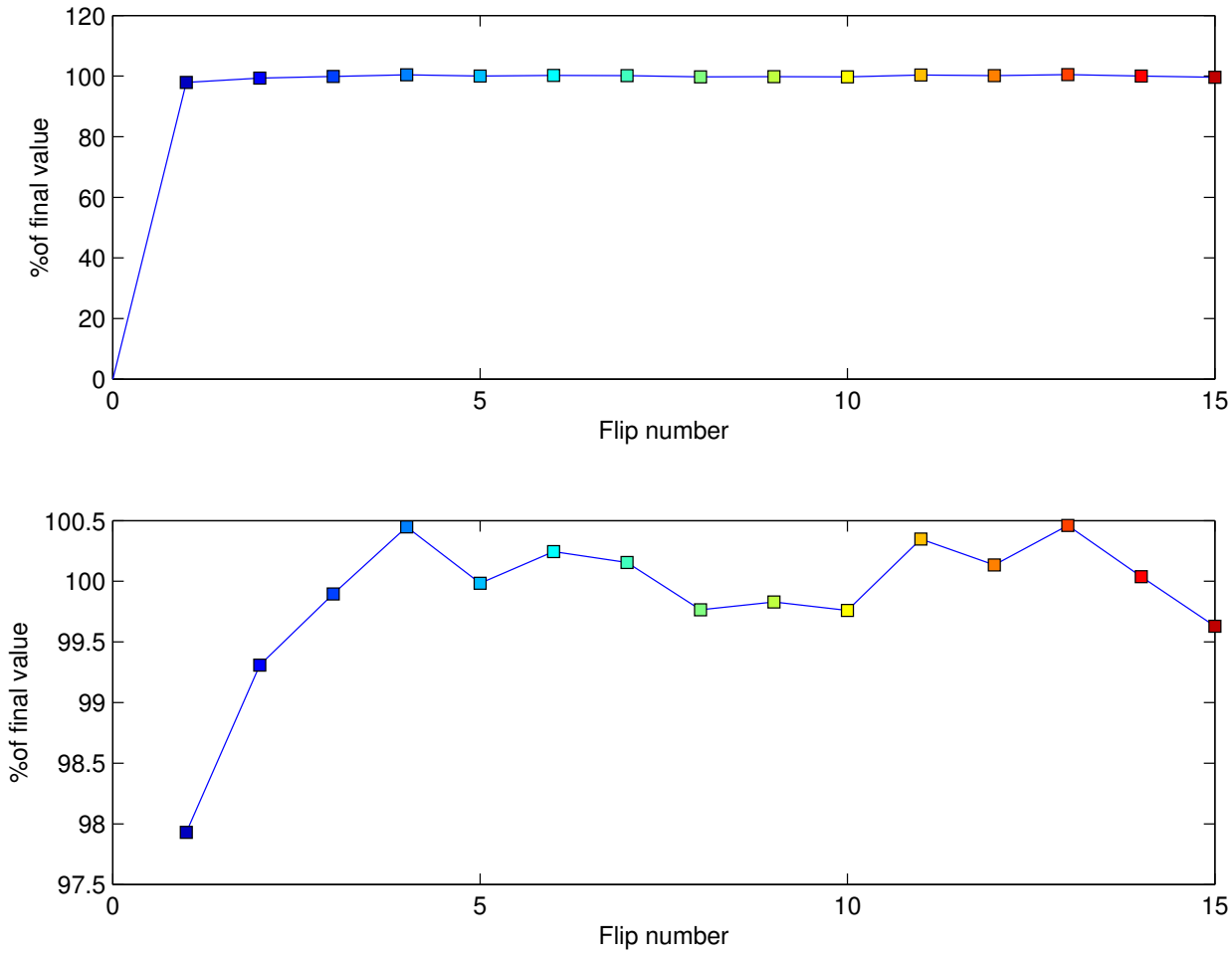


Figure 3.10. Torquer saturation

or failed component somewhere. This information is logged in flight and can be downlinked to analyze if the torquers are firing correctly.

3.4 Sensors

The only sensor specified in [6] was a magnetometer. The magnetometer was proposed to be used to determine the angular rates of the ARC as well as an input to equation (2.2). Because the torque produced by the torquers depends on the Earth's magnetic field, the Earth's magnetic field must be known in order for the algorithm to compute the necessary dipole moment. The accuracy of the magnetometer was not specified but it needs to be able to determine rotation rates accurately enough for the ACDS to work.

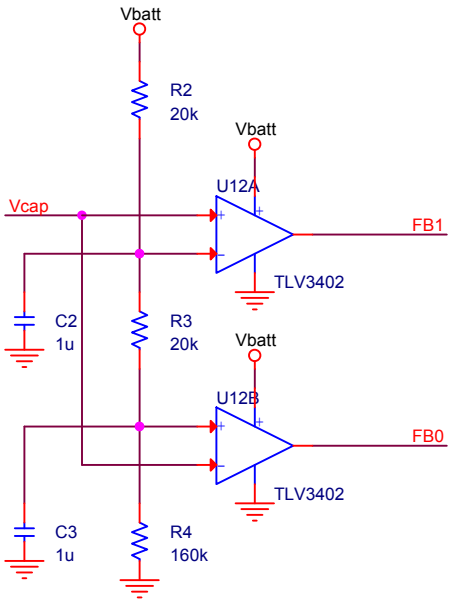


Figure 3.11. Torquer feedback comparators

3.4.1 Magnetometer

The magnetometers on the satellite are Honeywell HMC1052 Anisotropic MagnetoResistance (AMR) sensors. These sensors use AMR elements in a bridge configuration to measure the field in each axis. The HMC1052 is a 2-axis sensor that measures the field in the axes that are parallel to the board that it is mounted on (X and Y). Each face of ARC will have a single HMC1052 giving a total of 4 measurements in each axis.

The magnetometers are located on the back of the SPB, shown in figure 3.2. The magnetometer, with surrounding circuitry, is visible on the back of the board as are the Analog-to-Digital Converter (ADC) and amplifier. An accelerometer is also located on the back of the solar panel board and is used for a different mission objective. The data connector powers the sensors and provides access to the sensor data.

3.4.1.1 Magnetometer Amplifier

The SPB also contains an amplifier to amplify the magnetometer signal before it is read by the ADC. This allows the output voltage of the magnetometer to fill the input range of the ADC. The required range of the magnetometer will be largely dependent on the torquer geometry and could be more than the ±600 mGauss or so of Earth's field. The amplifier can be used to trade range for resolution depending on requirements.

Equation (3.1) shows the equation used to calculate the desired amplifier gain. The maximum

voltage readable by the ADC, V_{adc}, is divided by the maximum expected output from the magnetometer. The maximum output from the magnetometer is found by adding the output due to the desired full scale field and the maximum possible bridge offset,

$$A = \frac{V_{adc}}{B_{max} \cdot V_{bridge} \cdot S_s + V_{os} \cdot V_{bridge}} \,. \tag{3.1}$$

Equation (3.2) shows the gain calculations for the magnetometer amplifier to measure a ±4 Gauss field. The bridge and ADC reference voltages are 3.3 V. This gives an input voltage range for the ADC of ±1.65 V,

$$\begin{aligned} A &= \frac{1.65 \text{ V}}{4 \text{ Gauss} \cdot 3.3 \text{ V} \cdot 1 \text{ mV/V/Gauss} + 1.25 \text{ mV/V} \cdot 3.3 \text{ V}} \\ &= \frac{1.65 \text{ V}}{13.2 \text{ mV} + 4.125 \text{ mV}} \\ &= \frac{1.65 \text{ V}}{17.325 \text{ mV}} \\ &= 95.24 \,. \end{aligned} \tag{3.2}$$

3.4.1.2 Magnetometer Analog-to-Digital Converter

Each magnetometer has its own ADC on each SPB. The ADC used is the LTC2487. The LTC2487 is a 16-bit delta sigma ADC that has two differential analog input channels and an I2C interface. For a bridge voltage of 3.3 V, the HMC1052 has a sensitivity 3.3 mV/Gauss. If the amplifier gain is 95 this results in a sensitivity of 0.31 V/Gauss. For a 3.3 V reference voltage, the ADC has a resolution of 25 μV, which results in a resolution of 81 μGauss / LSB.

3.4.1.3 Magnetometer Operation

The HMC1052 sensors used on the ACDS are AMR sensors. They work by having a bridge with 4 elements that all change resistance with the applied magnetic field. The AMR sensors are primarily sensitive to magnetic fields in their sensitive direction, (H_s), but they are also slightly sensitive to magnetic fields in an axis normal to the sensitive direction, (H_C), called the cross axis. The sensor is only sensitive to magnetic fields that are in the film plane of the AMR sensor. For the 2-axis HMC1052 this means that both sensors show some sensitivity to fields in both the X and Y axes as shown in figure 3.12. The cross axis effect varies from sensor to sensor so calibration values must be calculated for each sensor [14].

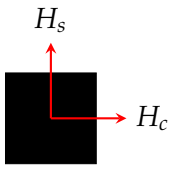


Figure 3.12. Magnetometer showing H_s and H_c

Equation (3.3) shows the simplified equation for the magnetometer as described in [14]. V_b is the bridge voltage that is applied to the sensor. S_s is the sensitivity in the sensitive direction. D is the cross field sensitivity offset. The HMC1052 datasheet[15] mentions that the sensors have a bridge offset. The bridge offset is the sensor output when zero field is applied to the sensor and can be as large as ±1.25 Gauss. The bridge offset is represented by V_{os} in equation (3.3),

$$V_s = V_b \left(S_s H_s + D \cdot H_c + V_{os}\right) . \tag{3.3}$$

The bridge voltages are measured using an ADC that uses the bridge voltage as its reference. To take this into account and simplify the equations, the following substitution is made:

$$V_s' = \frac{V_s}{V_b} . \tag{3.4}$$

Equation (3.3) is useful to determine the voltage that would be produced for a given magnetic field condition, but not the reverse when the sensor voltages are known but the fields are not.

Because the calculated field in the sensitive direction also depends on the field in the cross axis direction, equation (3.3) was duplicated for the cross field sensor voltage and both equations were solved simultaneously. This resulted in twice as many constants as equation (3.3). The voltages V_s' and V_c' are the normalized ADC voltages in the sensitive and cross axes, respectively. Each of the constants from equation (3.3) has similar duplicate forms that comes from the extra equation that was used to solve for H_s. To distinguish the constants additional subscripts have been added. For example S_{ss} is the sensistivity of the sensitive axis field to the sensitive axis input while S_{sc} is the sensitivity of the cross axis to magnetic fields in the sensitive axis direction. Equation (3.5) shows equation (3.3) solved for the unknown quantity, H_s,

$$\begin{aligned} H_s = & \frac{V_s'}{S_{ss} - \frac{D_s \cdot D_c}{S_{sc}}} - \frac{D_s \cdot V_c'}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\ & - \frac{S_{sc} \cdot V_{oss} - D_s \cdot V_{osc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} . \end{aligned} \tag{3.5}$$

Equation (3.5) can be used to calculate the magnetic field from measured voltages, but the relationship between the constants is complex. To simplify equation (3.5) the constants can be

consolidated, as shown in appendix B, to get equation (3.6),

$$\begin{aligned} H_s &= C_1 \cdot V_s' + C_2 \cdot V_c' + C_3\,, \\ H_c &= C_4 \cdot V_s' + C_5 \cdot V_c' + C_6\,. \end{aligned} \tag{3.6}$$

In this case V_s' and V_c' are the ADC values for the sensitive and cross axes, respectively. C_1, C_2, C_4, and C_5 are the sensitivity in the sensitive and cross directions and C_3 and C_6 correct for the bridge offset of the sensor and also compensates for external offsets. These offset values are used to correct for the torquer offsets.

3.4.1.4 Magnetometer Calibration

To calibrate the magnetometer, the coefficients in equation (3.6) must be calculated. The magnetometer is first placed inside the Helmholtz Cage. The field is set under MATLAB control so the entire calibration process can run automatically. The calibration program sweeps the field inside the Helmholtz Cage through a predefined sequence and reads the sensor output at each point[16].

The method of least squares is used to solve equation (3.7) for the coefficients in equation (3.6). Each line in the $\boldsymbol{A}$ matrix represents a separate magnetic field measurement. H_n is the n^{th} value set by the Helmholtz cage in the sensitive axis,

$$\overrightarrow{b} = \boldsymbol{A}\overrightarrow{x}$$

where

$$\begin{aligned} \overrightarrow{b} &= \begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix} \\ \boldsymbol{A} &= \begin{bmatrix} V_{s1} & V_{c1} & 1 \\ \vdots & \vdots & \vdots \\ V_{sn} & V_{cn} & 1 \end{bmatrix} \\ \overrightarrow{x} &= \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}. \end{aligned} \tag{3.7}$$

The least squared solution minimizes the error across all data points. This results in calibration values that minimize error across the range of calibration points that were taken. This means that the points used to do the calibration should span the range of values that the magnetometer is expected to measure.

3.4.2 MEMS Gyros

For flight, a LPY410AL MEMS gyro will also be used to sense rotation rates. MEMS gyros have limited resolution compared to what can be achieved with the magnetometer. The LPY410AL has a maximum range of ±400°/sec. The LPY410AL has a zero-rate sensitivity drift of 0.03 dps/°C. The desired rotation rate of the satellite is about 500 μdps. This means that for a 1°C temperature change the rotation rate could change by 58 times the desired rotation rate. The LPY410AL also has a rate noise density of 0.014 dps/ $\sqrt{\text{Hz}}$ so even at a slow sample rate of 1 Hz, the noise is 27,000 times larger than the desired rotation rate.

MEMS gyros do, however, give good results if the rotation rates are large, such as after the satellite is ejected from the P-POD. If the satellite is rotating fast enough, the magnetometer readings will alias and it can appear that the satellite is rotating at a speed that is much less than the actual speed. If the rotation rates are too high for the MEMS gyro, the output will saturate, and while the rotation rates are not measurable, a direction and lower bound can be determined.

3.5 Embedded System

At the heart of the ACDS is an embedded system. This system is responsible for running the algorithm, taking housekeeping data and, interfacing with the rest of the satellite.

3.5.1 Processor

The processor used for the ACDS and all systems on ARC is the Texas Instruments MSP430F2618. The MSP430 is a microcontroller aimed at low power applications. The MSP430f2618 runs at a maximum speed of 16 MHz and has 8 kB of Random Access Memory (RAM) and 116 kB of flash memory. The MSP430 microcontrollers do not have an external memory bus so there is no way to add extra RAM or flash memory to the address space of the MSP430.

3.5.2 SD card

For flight data storage a SD card is used. This is necessary because of the needed space and write times. The internal flash is limited and also needs to be used for program storage making it undesirable for data storage. Furthermore, the internal flash can't be read while it is being written. Write times for the internal flash are fairly slow and require that the processor disable interrupts, making normal operation difficult. The flash shares the same address space as the RAM and has the same access time. This makes the internal flash an attractive place to put settings and calibration data that are written only a few times but accessed often as any data on the SD card must be copied into RAM before being used.

3.5.3 ARC Bus Communication

The sensor data is read by the LEDL board and sent over the bus to the ACDS board. The ACDS board initiates the connection by sending a command to tell how often measurements should be sent. The LEDL then takes sensor data and sends it at the requested interval.

Chapter 4
Helmholtz Cage

The testing and calibration of the ACDS requires a controllable, uniform magnetic field. The Helmholtz cage was constructed in 2009 by the ARC mechanical team. The cage was designed to be able to create a 2 gauss magnetic field in any direction within the cage. This allows the Earth's magnetic field to be canceled out and the field inside to be equivalent to the Earth's field at any point above the Earth's surface.

4.1 Hardware

The Helmholtz cage consists of 3 pairs of perpendicular coils as shown in figure 4.1. Each coil pair generates a nearly uniform magnetic field perpendicular to the plane of the coil. Together the three perpendicular coil pairs can generate a field in any direction.

Figure 4.1. The Helmholtz cage used for ACDS testing

The coils are driven by a set of six Agilent E3633A power supplies as shown in figure 4.2. The power supplies have a maximum current of 20 A for output voltages less than 8 V or a maximum current of 10 A for output voltages of 20 V. The supplies are computer controlled with a General Purpose Interface Bus (GPIB) connection.

Both coils in each pair require the same current to be driven in each coil to get a uniform field. This could be achieved by putting both coils in a pair in series but this would make the required

voltage too large to be feasible. The coils could also be put in parallel but this requires the coils to be matched. The Helmholtz cage software sets the power supply currents to be the same for both coils in each pair.

An Alpha Lab #3AMG Milligauss Meter is used to measure the magnetic field of the Helmholtz cage. The meter is visible in figure 4.2 to the left of the computer monitors. The Milligauss Meter has a digital display and an analog voltage output. The voltage output is read by a NI PCI-6010 Multifunction Data AQuisition (DAQ) installed in the Helmholtz cage computer.

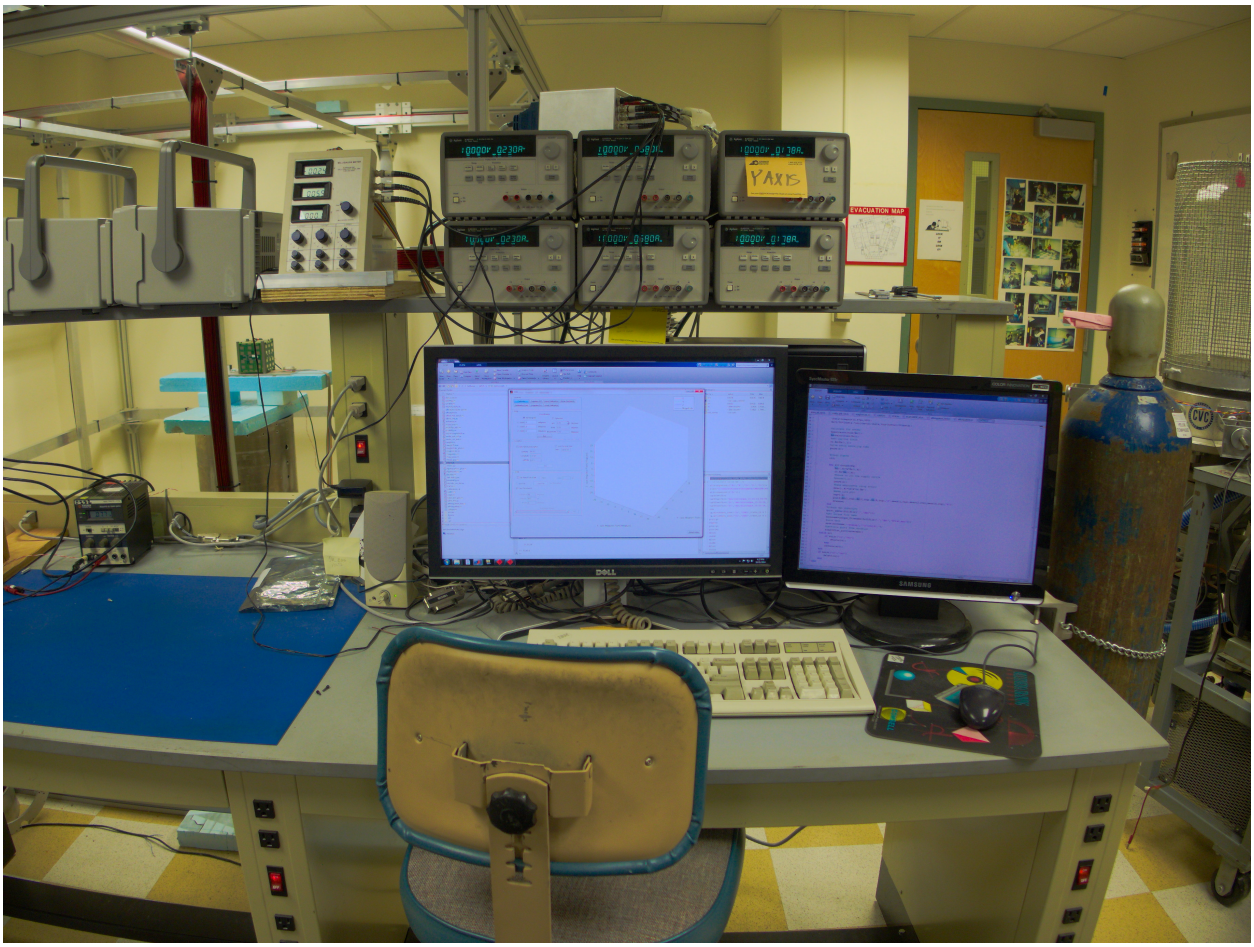


Figure 4.2. The Helmholtz cage computer and power supplies

4.2 Software

Figure 4.3 shows the block diagram of the Helmholtz cage. The cage control class is a MATLAB class that is used for low level interfacing to the Helmholtz cage hardware. The GPIB interface is used to connect to the power supplies. The power supplies are connected to the coils through the H-bridge so that the current direction can be reversed. The H-bridge is controlled using the digital

I/O on the DAQ card. The magnetometer is connected to the analog input channels of the DAQ and reads the magnetic field in the Helmholtz cage.

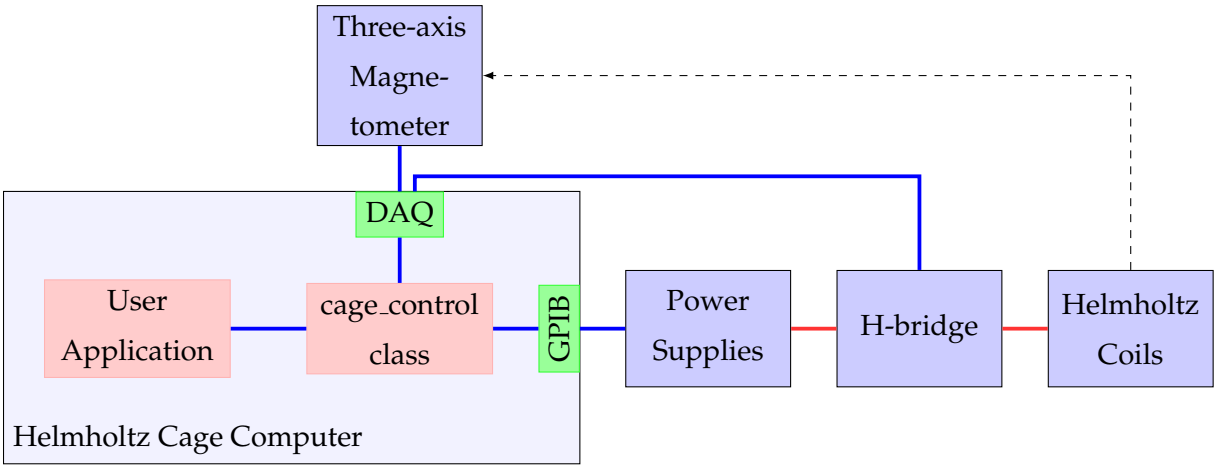


Figure 4.3. Helmholtz cage block diagram

The feedback from the magnetometer is used to calibrate the Helmholtz cage. The cage is calibrated by setting the power supplies to a range of currents and reading back the magnetic field. The data is then used to compute a 4×4 transformation matrix, `T`, to determine the current required to generate a given magnetic field. The magnetometer is not used in a closed loop fashion because magnetic fields produced by a device under test would change the magnetic field giving erroneous results.

The cage control class takes care of all of the low level hardware interaction and calibration for the Helmholtz cage. The cage control class opens interface objects to the power supplies and the DAQ. The Helmholtz cage is controlled by setting properties of the class. Setting the `Is` property changes the current that the power supplies drive through the coils. The `Bs` property is a dependant property. Setting the `Bs` property calculates the required currents using the calibration matrix `T`, and sets `Is`. Reading the `Bs` property calculates the magnetic field setpoint using `I` and `T`. The read only `Bm` property is also dependant as it represents the magnetic field read by the sensor. Reading `Bm` causes the output voltage of the magnetometer to be read by the DAQ and converted to a magnetic field reading.

4.2.1 Compatibility

The cage_control class was originally developed on a 32-bit machine. When the code was moved to a 64-bit machine running MATLAB R2011b the code did not work. This was because the cage_control class uses the legacy interface for the DAQ which is not supported in 64-bit MATLAB.

The 64-bit version of MATLAB only supports the session-based interface which is the newer interface. The NI PCI-6010 is not usable with the session-based interface in MATLAB R2011b to R2013b due to a known bug. The work around was to install 32-bit MATLAB on the 64-bit machine which allowed the legacy interface to be used. The bug is supposedly fixed in MATLAB R2014a so the cage_control class could be ported to the session-based interface in the future.

4.3 Testing

The accuracy of the Helmholtz cage is important because it is used to measure the accuracy of the magnetometers on the CubeSat. The Helmholtz cage is not magnetically shielded so outside disturbances can affect the magnetic field in the cage.

To test the Helmholtz cage the commanded magnetic field was swept as shown in figure 4.4 and the field inside the cage was measured with the Alpha Lab #3AMG Milligauss Meter. The measured field follows the commanded field for the most part but there are several areas where there is a visible difference between the measured and commanded field.

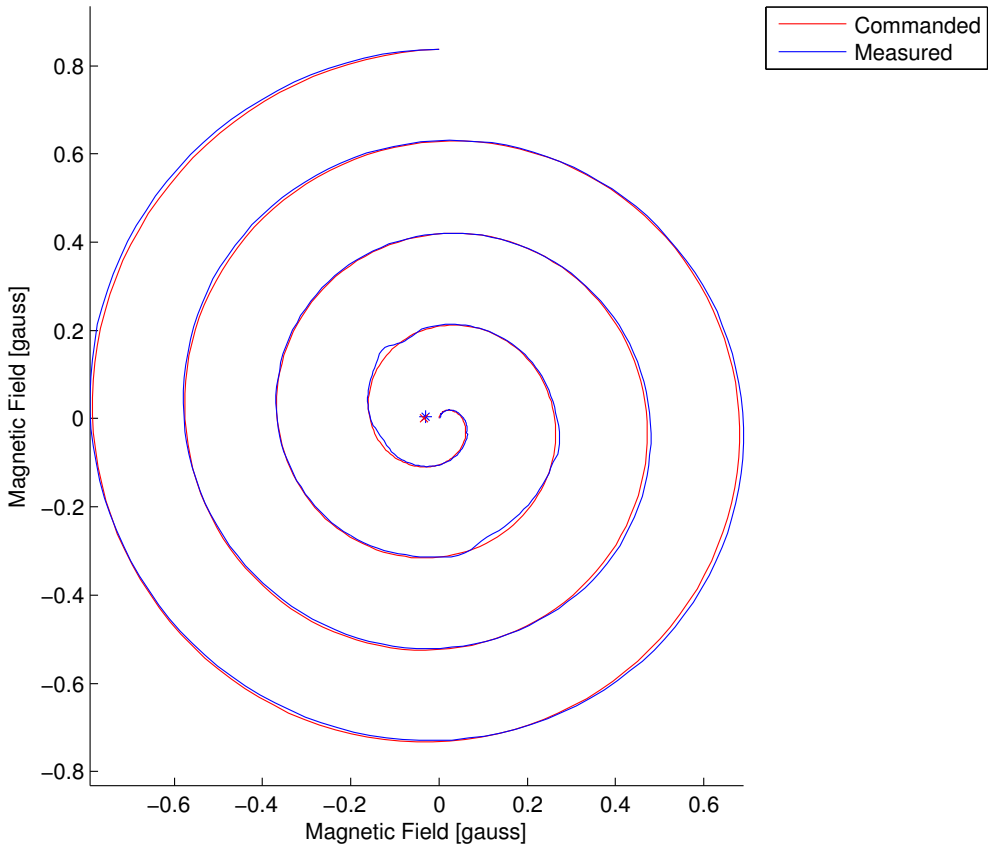


Figure 4.4. Helmholtz cage test field

Figure 4.5 shows the magnetic field error from the testing in figure 4.4. The total RMS error for the plot is 9.4 mGauss. There are a few large error spikes shown in the graph. Repeating the test shows that the errors are not consistent meaning that the errors are not caused by the cage itself but the environment around it. It is possible that the interference may be attributed to the position of the elevator or fire doors which are just outside the lab.

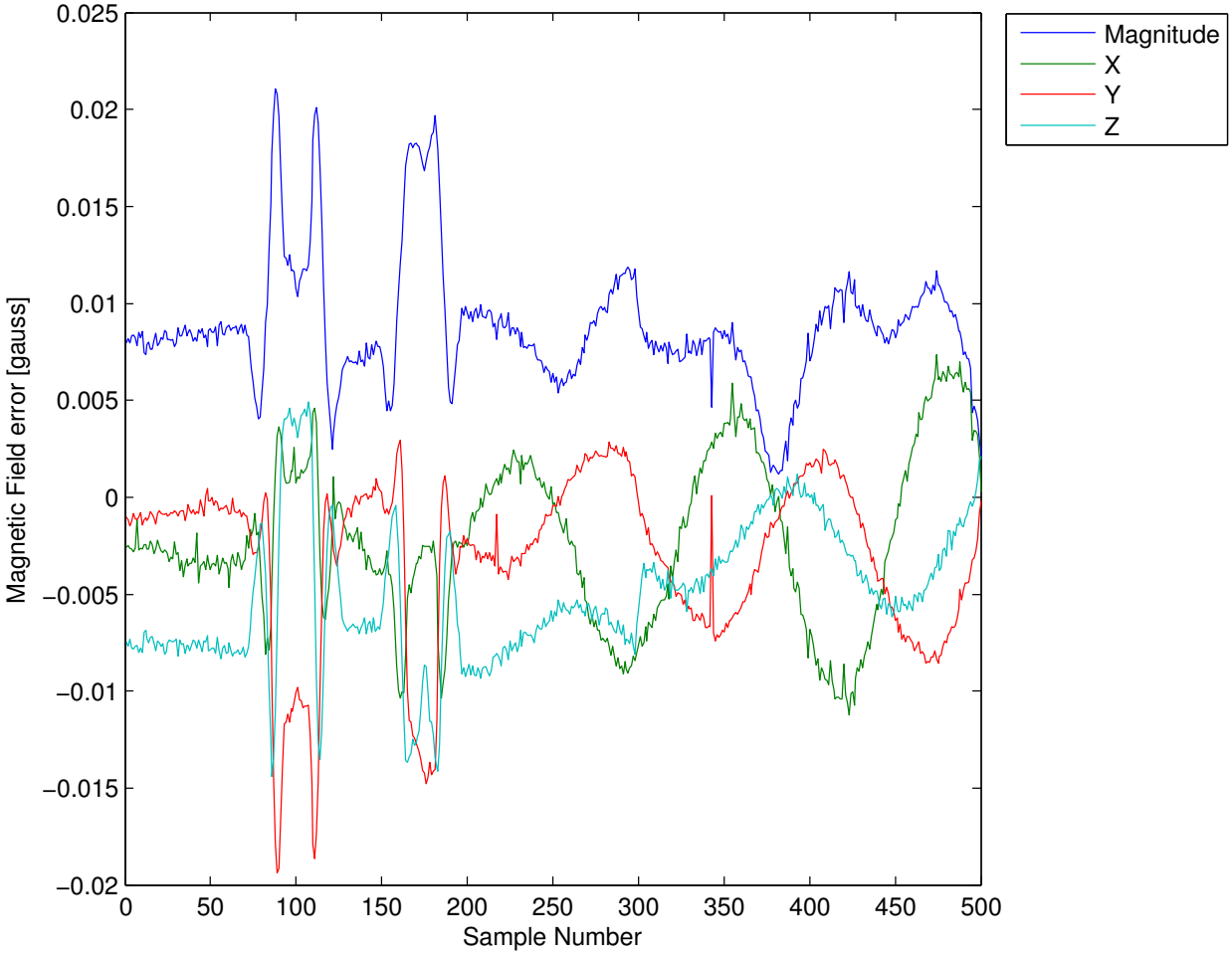


Figure 4.5. Helmholtz cage test error

Chapter 5
Software

The main responsibility of the ACDS software is to determine which torquer to flip each time step. The ACDS also needs to keep track of "house keeping" information, sensor readings, torquer states and internal control system states so that the performance of the algorithm can be tracked on the ground.

5.1 Overview

Figure 5.1 shows the overview of the ACDS and how it connects to other systems on the ARC. In order to determine which torquer to flip the ACDS needs sensor inputs. The sensors are read by the LEDL board and forwarded to the ACDS using the ARCbus. The ACDS needs to communicate with the COMM board to respond to ground station commands and downlink housekeeping data.

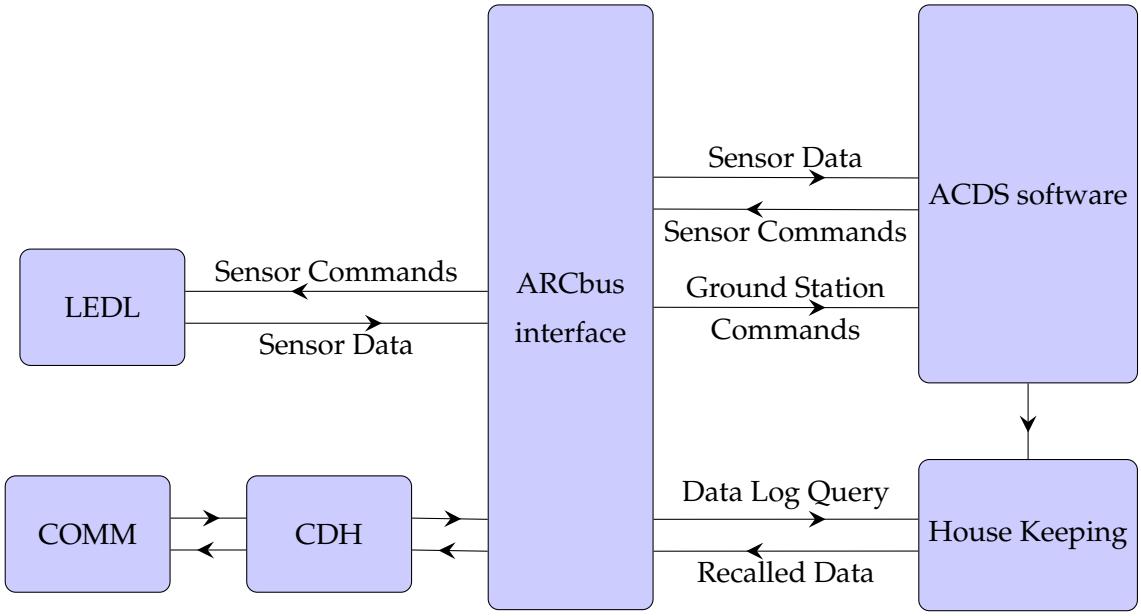


Figure 5.1. ACDS software overview

5.2 System Operations Overview

The ACDS starts running after the separation switch is switched and the power system applies power to all systems. Before starting any operations the ACDS waits for the "on" command from the Command and Data Handling (CDH) board. After the ACDS board receives the "on" command, the ACDS sends a command to the LEDL board to tell it to start taking sensor data. Once sensor data is received the ACDS starts to run the detumble algorithm.

5.2.1 Rotation Runaway

If the ACDS detects that the satellite is rotating too fast then it will automatically send the ACDS into a safe mode where the ACDS does not run. This could potentially happen for many reasons such as errors in the algorithm or unexpected conditions during flight. In this situation the best thing to do is for the ACDS to stop and wait for further intervention from the ground.

5.3 Ground Station Commands

Ground station commands will be provided to change the operation of the ACDS system in case it does not function as intended. Because the ACDS is an experimental system, it may run into unexpected problems. By uplinking commands some of the possible problems might be able to be diagnosed and fixed after ARC is in flight.

5.4 Algorithm Software

Figure 5.2 shows the software block diagram for the ACDS algorithm. Field measurements from the magnetometer are calibrated using the current torquer state and the compensation data. The B-dot algorithm uses the compensated data to calculate the dipole moments that should be generated. The dipole moments are then quantized as illustrated in figure 2.9. The algorithm decides which torquers to flip based on current torquer status and the desired torque. The torquer feedback checks to make sure that the torquers and drivers are functioning correctly. The torquer status is tracked by reading the torquer feedback and by knowing which torquers were flipped.

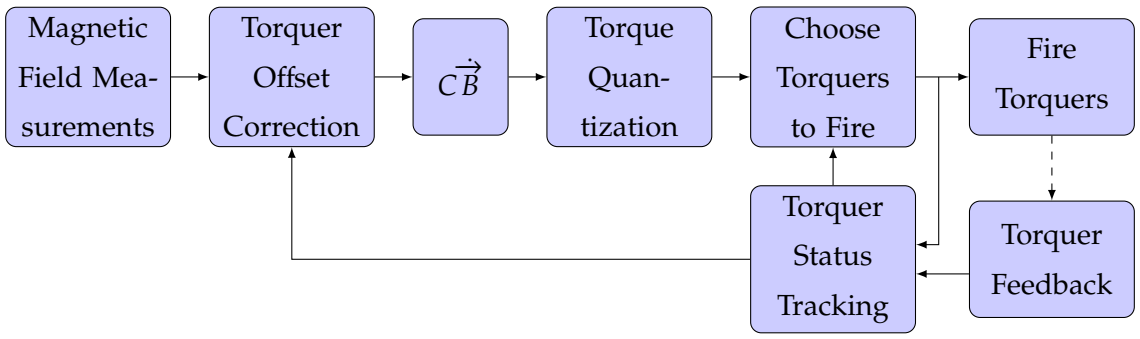


Figure 5.2. Overall Software Block Diagram

5.4.1 B-dot Algorithm

The B-dot algorithm is used to detumble the ARC. The B-dot algorithm uses the time derivative of the magnetic field, $\dot{\vec{B}}$, to calculate the magnetic dipole moment that the torquers should generate. This is a method that is widely used on other CubeSats that have magnetic ACDS[17]. In this

mode the magnetic dipole moment is simply set to a value that is proportional to the derivative of the magnetic field. The gain for the B-dot controller needs to be negative in order to oppose the motion of the magnetic field.

5.5 Auxiliary Software Functions

In addition to the algorithm software the ACDS also requires some auxiliary software functions. In order to generate the proper dipole moment the status of the torquers must be tracked in software so that the correct torquer can be flipped. The torquer status also needs to be tracked so that the torquer offsets can be subtracted from the magnetometer measurements.

5.5.1 Torquer Flipping Logic

The torquer flipping logic keeps track of the torquer status and sets the torquers to the dipole moment that is closest to the dipole moment requested by the control algorithm. The torquer flipping logic also attempts to distribute the flips somewhat evenly among the torquers in any given axis. This way if there are any degradation effects in the torquer field or the hardware, they will be minimized.

5.5.1.1 Status Tracking

The status of each torquer is tracked by the flipping logic. The status information includes direction, initialization, and error flags for each torquer. The last four torquers that were flipped are also tracked. This is used to choose which torquer to flip.

When it is time to flip a torquer the software looks at the current torquer status and the desired torque to figure out which direction a torquer needs to be flipped. Next the software finds torquers that can be flipped in the needed direction. Finally, the torquer that was flipped the least recently is flipped. This way the torquer flips are distributed across the torquers in a given axis. If torquers are marked as uninitialized then they are flipped first before other torquers so that all torquers are in a known state as quickly as possible after a reset or power-up.

5.5.1.2 Torquer Feedback

Before and after each torquer flip the torquer feedback comparators are read. After the torquer flip is complete the feedback values are examined to determine if the torquer flipped. If the capacitor voltage was below the upper threshold before the flip then an error is flagged in the torquer status indicating that there is a problem with the capacitor. If the capacitor voltage after a flip is not below the lower threshold then there is likely a bad connection to the torquer windings. Torquers

that failed to flip are flagged and not flipped in the future. It is possible that the comparator could be broken and indicate that it is both above the upper threshold and below the lower threshold. In this case a flag is set to indicate the error and the torquer is assumed to have flipped as normal.

5.5.2 Torquer Compensation

Because the geometry of the system should not change in flight, the field offset seen by each magnetometer depends on the combination of torquer states. By taking measurements in the Helmholtz cage these offsets can be calculated, as described in section 6.2.3, and eliminated during flight in order to measure the Earth's magnetic field.

5.5.2.1 Compensation Data Set

There are a total of 12 LPMTs on the ACDS. Each LPMT has two states which results in a total of 4096 possible states. If the offset for each state uses four bytes this results in a data set that is 16 kB. The field produced at each state is the sum of the field contributions of all of the torquers. The dataset can be reduced by doing three separate calibrations, one for the set of torquers in each axis. The calibrations are then combined by superposition to get a calibration set where one offset is chosen for each axis and added together to get the full offset. This reduces the number of states to $3 \times 16 + 1 = 49$, which is only about 200 bytes of data, much reduced from using all possible states.

5.5.2.2 Compensation Routine

To calculate the compensation values for the torquers, the calibration procedure described in section 3.4.1.4 is run for each combination of torquer states. The compensation routine calculates values for C_1, C_2, C_4 and C_5, which are the same calibration constants computed with the magnetometer calibration routine. The offsets are separated into a common pair of offsets and then 3 sets of 16 offsets for each set of torquers as described in section 6.2.3. The total offset value for a given torquer state is the sum of the common offset value and an offset determined by the state of the torquers in the X, Y and Z axes.

5.5.3 Data Logging

During ACDS operations data on the system status is collected during flight. This data is stored in nonvolatile memory and can be recalled and transmitted to the ground in order to evaluate the ACDS system performance. The data that is recorded is shown in table 5.1.

Table 5.1. ACDS operations data format

Data	size (bytes)	Format
Time Stamp	4	unsigned integer
Mode	1	unsigned integer
Flags	1	flags
Torquer feedback	2	flags
Magnetic flux vector	12	floating point vector
M_{cmd}	12	floating point vector
Torquer status	12	flags
Torquers flipped	6	unsigned integer (one per axis)
Raw magnetometer readings	24	unsigned integer (one per axis)
Gyro readings	6	unsigned integers (one per axis)
B-dot	12	floating point vector
Total :	92	

5.5.4 On-Board Data Processing

Because the downlink data speed is limited, it is necessary to reduce the data that needs to be downlinked. One way to do this is to through on-board processing to reduce the data before it is downlinked. For the ACDS system this will most likely take the form of returning only the desired data from the recorded data set or returning min/max values from within a data set.

5.5.5 Beacon Data

The ARC COMM system sends a beacon packet every 10 seconds. The purpose of the beacon is to be able to locate the ARC and determine how well the satellite is functioning. The beacon packet contains a small amount of data from each subsystem which gives an idea of how well the system is operating.

The beacon data from the ACDS is shown in table 5.2. The data contains enough data to be able to see how well the system is operating but not exceed the maximum packet size. As discused in section 7.2 the Kalman filter may be added as a future enhancement. The Kalman filter attitude field of the beacon packet will be filled with zeros until this is written. Similarly the Kalman filter rates will be filled with the gyro rotation rates, until the Kalman filter is implemented.

Table 5.2. ACDS Beacon Data format

Data	size (bytes)	Format
Current magnetometer readings	6	signed integers (one per axis)
Mode	1	unsigned integer
Current torquer status	3	flags
Number of torquer flips so far	6	unsigned integers (one per axis)
Kalman filter attitude	8	integer quaternion
Kalman filter rates	6	integer vector
Total :	30	

Chapter 6
Verification

The ACDS is a unique system that was developed specifically for the ARC and has not yet been flight tested. Ideally everything would be verified before launch, but it is difficult to replicate the on-orbit environment closely. The magnetic field can be effectively simulated using the Helmholtz cage so the sensors can be calibrated and tested.

Using the Helmholtz cage the ACDS system can't be fully tested. The algorithm will flip torquers if a rotating magnetic field is generated, but this only tests how the algorithm responds to the magnetic field inputs but does not include the effect the torquers have on spacecraft dynamics.

6.1 Torquer Offsets and Repeatability

The LPMTs used for the ACDS pose a unique challenge when compared to both passive magnetic torquers and active proportional torquers. The LPMTs can be changed like active proportional torquers but, they can't be switched off to produce zero field. Passive torquers use permanent magnets that only opperate in one state and are easily calibrated out, but the LPMTs can be in many states. Magnetometer measurements are used for attitude determination but the field from the torquers, if not compensated for, will overwhelm the measurements of the Earth's magnetic field making attitude determination impossible.

6.1.1 Torquer Offsets

The magnetic field of the Earth ranges from 0.3 to 0.6 Gauss [18, pp. 114]. In order to measure the attitude of the ARC to within ±5°, the Earth's magnetic field must be measured with an error of less than 26 mGauss. This means that the offsets from the torquers must be known with an accuracy of at least 26 mGauss.

Because of the nature of the dipole magnetic field, the offsets measured by each magnetometer are unique to the relative placement between each magnetometer and torquer. Some torquers have only a small effect on each magnetometer and some have a much larger effect.

The torquer compensation data, explained in section 6.2.2, are used to show the effect that the torquers have on the magnetometers. The `offsetplot` function is used to plot the changes in torquer offset. The offset part of the compensation data is plotted with respect to torquer state.

Figure 6.1 shows the plot from `offsetplot`. The plot only shows the 48 torquer states used for calibration out of the 4096 possible states. The state of each torquer is indicated with a "+" if the torquer was flipped in the positive direction and a "-" if the torquer was flipped in the negative direction. The torquers are separated into groups of four torquers by axis. For the locations of torquers on the ACDS board refer back to figure 3.6. The first group is for the X-axis and the last is

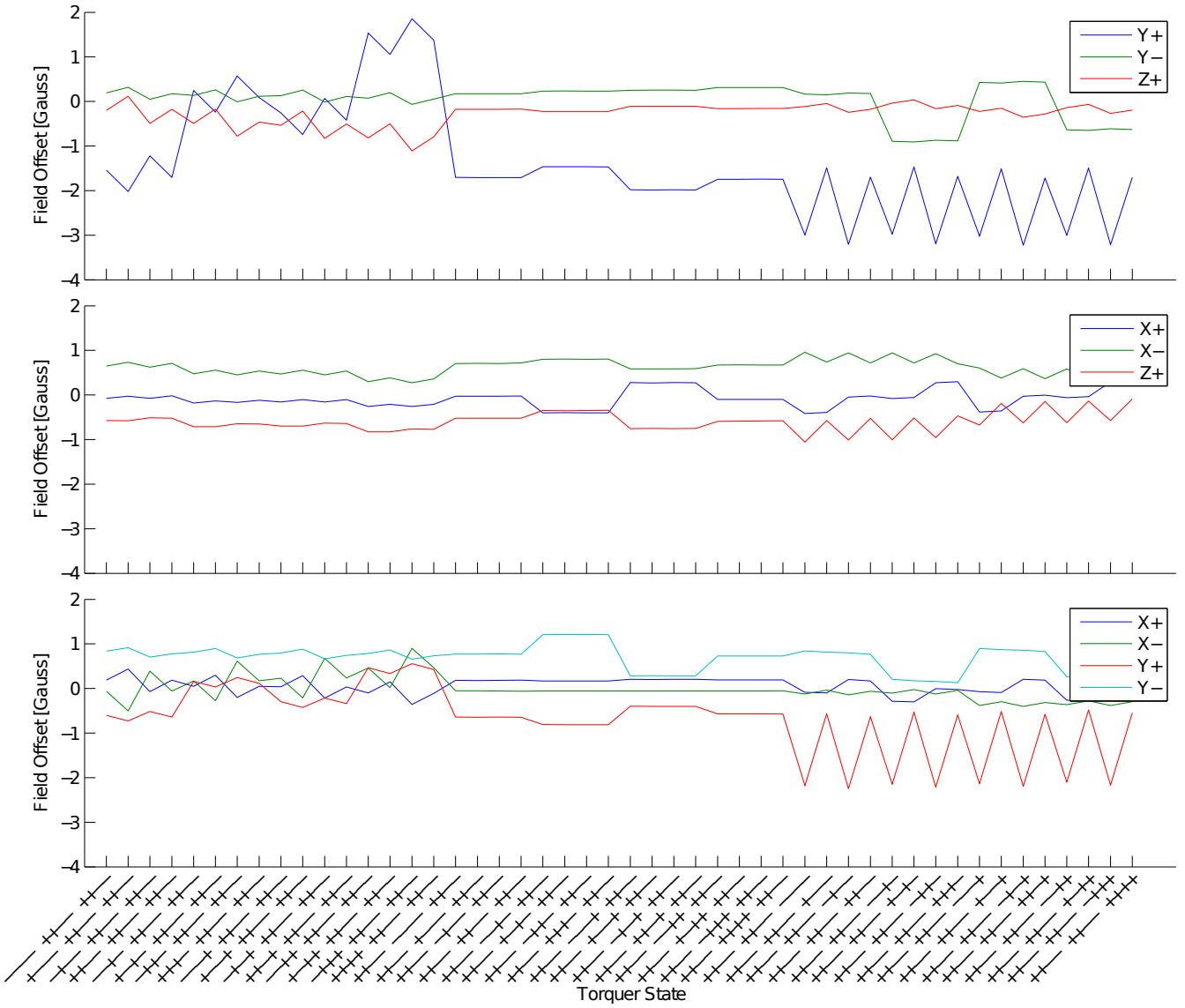


Figure 6.1. Magnetic field offsets due to changing torquer states as measured by different SPBs

for the Z-axis. In the first 16 states the X-axis torquers are flipped in all possible states, for the next 16 the Y-axis torquers are flipped through all possible states and for the last 16 states the Z-axis torquers are flipped through all possible states.

Table 6.1. Information from `offsetplot`. All numbers are in Gauss

SPB	Max Offset	Min Offset	Deviation
X-axis			
Y+	1.85	-3.23	5.08
Y-	0.45	-0.91	1.36
Z+	0.12	-1.11	1.23
Y-axis			
X+	0.32	-0.42	0.73
X-	0.96	0.28	0.68
Z+	-0.09	-1.06	0.97
Z-axis			
X+	0.44	-0.36	0.79
X-	0.90	-0.50	1.41
Y+	0.55	-2.25	2.80
Y-	1.21	0.13	1.08

Table 6.1 shows the offset maximums and minimums. The deviation is the difference between the maximum and minimum torquer offsets for a given magnetometer. The deviation is an indication of how much of an effect the torquers have on the magnetometers. The greatest deviation is just over 5 Gauss for the Y+ SPB measuring the X-axis. The Y+ SPB also has the second highest deviation of almost 3 Gauss. The Y- and Z+ SPBs have deviations that are closer to 1 Gauss. Looking back at figure 3.5 it makes sense that the Y+ SPB has the largest torquer offset as it is located close to a pair of X torquers and a pair of Z-torquers. The Y- SPB is located next to the header so the torquers are located farther away. The Z+ board is located farthest away from the torquers so one would expect that the offsets for it would be smaller.

Figure 6.1 and table 6.1 show the necessity of torquer compensation. There are large variations in the measured magnetic field due to the torquers changing state. These variations in the magnetic field measurements make attitude determination impossible without torquer compensation. What figure 6.1 and table 6.1 do not show is how repeatable the offsets are which determines if compensation is possible.

6.1.2 Torquer Repeatability

The torquer compensation method depends on the torquer offsets being repeatable. If the offsets are not nearly repeatable then the offsets will induce errors in the measured field. Figures 6.2 and 6.3 each show a plot of the torquer offsets. The plot shows the offsets of the torquers when they are polarized in each direction. Each plot shows a different magnetometer axis. The offset values are normalized to show deviation from the median for better comparison. The maximum offset variation in figures 6.2 and 6.3 is 20 mGauss.

The plot in figures 6.2 and 6.3 were generated using the `alltorquetest` function. The plots for the Y+ SPB are shown because, as shown in table 6.1, the Y+ board has the largest torquer offsets. To create the plot each torquer was flipped 20 times in each direction. After each flip a magnetometer calibration was done to get the offset values. The variation in the offset is then plotted as a box plot.

Figures 6.2 and 6.3 were plotted using MATLAB's `boxplot` function. The boxes extend from the 25^{th} to the 75^{th} percentile. The whiskers extend to the most extreme point in each direction that is not an outlier. Outliers are those points that are larger than $q_3 + 1.5\,(q_3 - q_1)$ or smaller than $q_1 - 1.5\,(q_3 - q_1)$ with q_1 and q_3 the 25^{th} and 75^{th} percentile respectively. This corresponds to points outside roughly $\pm 2\sigma$ for normally distributed data. [19]

The plot in figures 6.2 and 6.3 has the `'extrememode'` option set to `'compress'` with `'datalim'` set to ±20 mGauss. This prevents any points outside ±20 mGauss from compressing the scale of the plot so that the data of interest is easier to read. The dotted line marks the limit of the limit set by `'datalim'` and the gray lines mark the compression region. [19]

Figures 6.2 and 6.3 show that there is some variation in the torquer offsets. The variations, except for outliers, are all below ±15 mGauss. Data was taken for the X+, X-, Y+, Y-, and Z+ SPBs. All of the plots are shown in appendix A. The variation in the other SPBs is similar to the Y+ except a few SPBs show slightly smaller variations. Data was not taken for the Z- board because the test setup does not have a way to connect to it. Furthermore the ACDS board is near the center of the CubeSat and the torquer locations are similar when viewed from the Z+ and the Z- faces.

6.2 Magnetometer Verification

There are several steps to verify and calibrate the magnetometers on the SPBs. First the board is tested separately from the rest of the CubeSat. Next the SPB is attached to the CubeSat and the torquer compensation routine is run to calculate the torquer offsets. Finally the compensation values are transfered to the ACDS board and the compensation is verified.

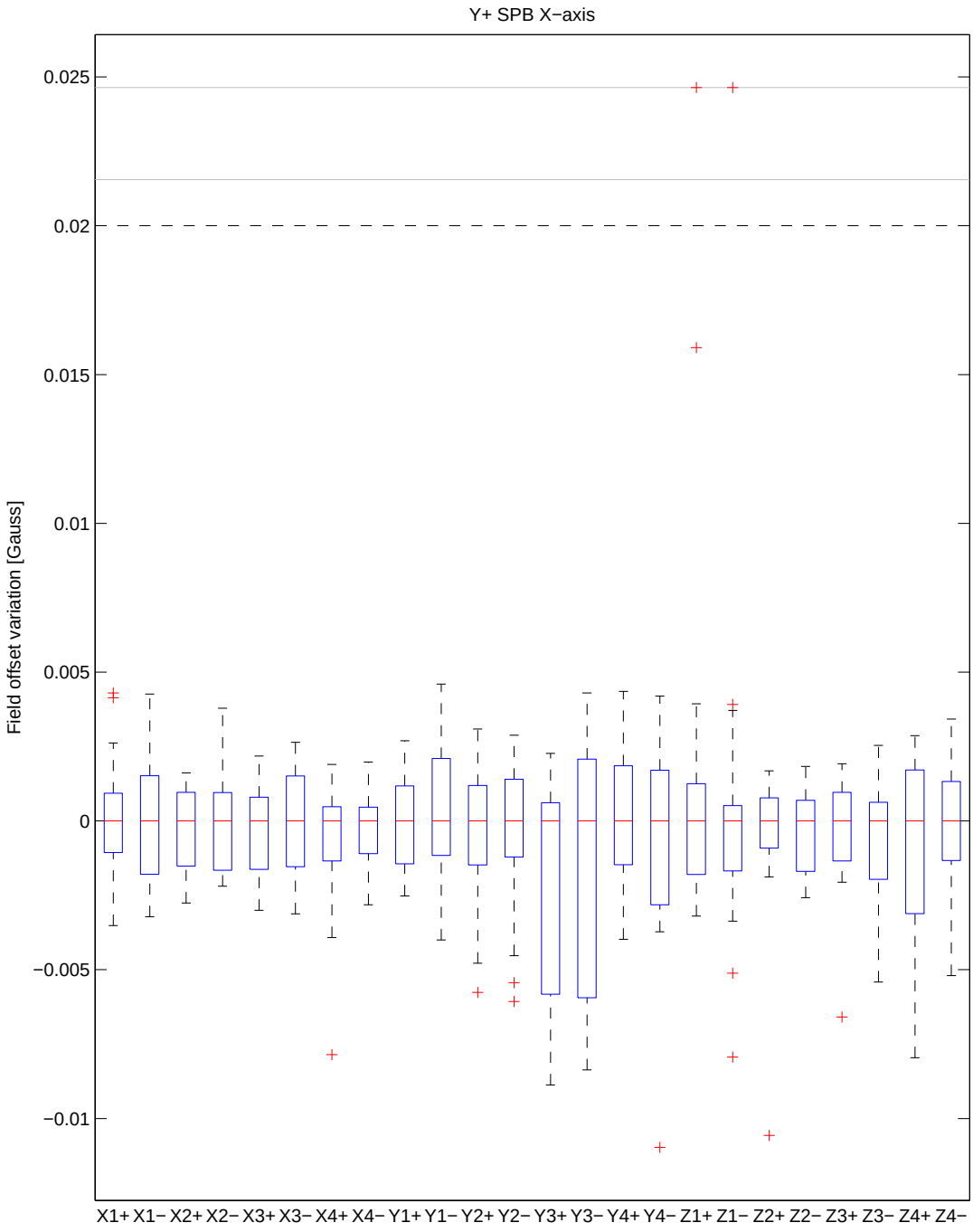


Figure 6.2. Box plot showing torquer offset variations for the X-axis of the Y+ SPB

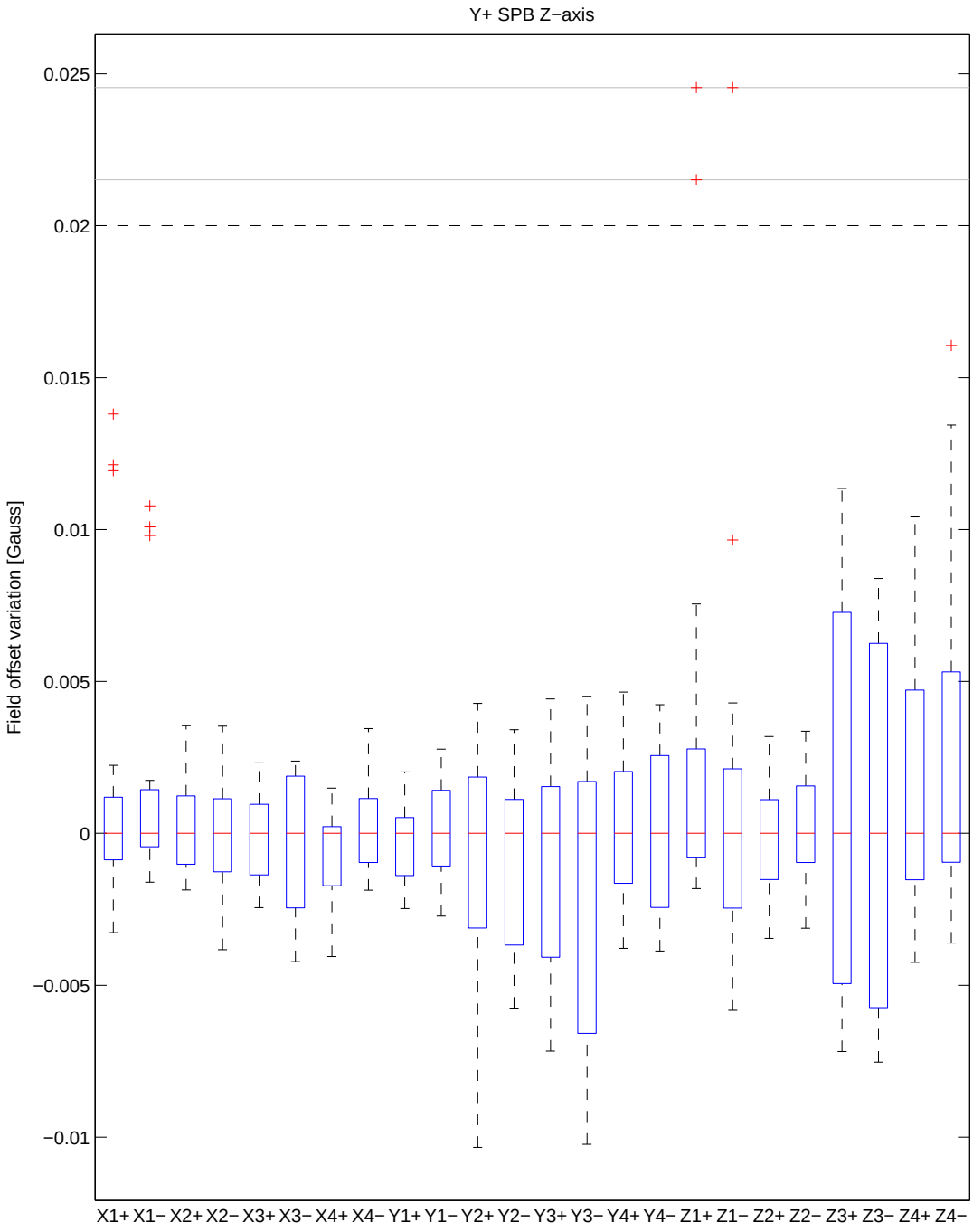


Figure 6.3. Box plot showing torquer offset variations for the Z-axis of the Y+ SPB

6.2.1 Initial Verification

The magnetometer verification is done using the Helmholtz cage. The verification is used to test a SPB to see if the magnetometer is performing correctly. The `spbMag` function is used for verification. A calibration (see section 3.4.1.4) is run on a board separate from the CubeSat. The calibration coefficients are used to calculate parameters found on the datasheet to verify that the magnetometer is performing within specifications.

Listing 6.1. SPB verification results

```
X  Ss     0.9628 mV/V/Gauss    Pass
X  Ds    -7.9555%              Pass
X Vos    -0.1220 mV/V          Pass
Y  Ss     1.0128 mV/V/Gauss    Pass
Y  Ds    -5.1552%              Pass
Y Vos    -0.1989 mV/V          Pass
All Tests Passed!!!
```

Listing 6.1 shows the output from the SPB verification. The calculated datasheet parameters are shown along with pass or fail depending on if the values are within the maximums specified by the datasheet.

6.2.2 Torquer Compensation

The `calall` function is used to calculate the torquer compensation data for a SPB and the full set of torquers. The `callall` function calls the `tCal` function three times, once each for the X, Y and Z axis. The `tCal` function works similarly to the magnetometer calibration, but the field is swept through the sequence 16 times, once for each torquer state. The data from each sweep is used to calculate the scaling factors for the compensation (C_1, C_2, C_4, and C_5 from equation (3.6)) for all torquer states as well as offset values, C_3 and C_6 from equation (3.6), for each torquer state.

The calibration values for each torquer axis are computed separately. This gives three sets of scaling factors and 3 sets of 16 offsets which must be combined into a single data set that can calculate the offset for any torquer state. The offset due to a particular torquer state is the sum of the offset from the magnetometer and the offset caused by each torquer. The "`++-- ++-- ++--`" state is common to the data from each axis. A common offset is calculated by averaging the offsets from the "`++-- ++-- ++--`" state. The common offset is saved and subtracted from the offsets for all states so that the torquer offset can be calculated as the sum of the common offset and one offset for each torquer axis. The three sets of scaling factors should be close to each other and are averaged together to get a single set of scaling factors for all states. The scaling factors, common

offset and offsets for each torquer state are combined together into a single compensation data set for each SPB.

6.2.3 Compensation Testing

The torquer compensation is tested using the `tCalTstFull` function. The function runs the Helmholtz Cage through a field sequence and takes a magnetometer measurement at each point in the field sequence. During the process a random torquer is flipped every 10 field sequence points. The measurements are corrected, in MATLAB, using the compensation data set and compared to the expected field. The RMS error is also calculated for the data set so the effectiveness of the torquer compensation can be compared to the calibration without torquers.

Figure 6.4 shows a plot of the Y+ compensation test data. The "commanded" values are the magnetic field values that were programmed into the Helmholtz cage. The "scale only corrected" values are the measurements from the magnetometer corrected using the scaling values (C_1 C_2 C_4 and C_5 from equation (3.6)) from the correction but not the offset values (C_3 and C_6 from equation (3.6)). This shows what could be achieved without torquer compensation. The "corrected" values are the measurements from the magnetometer corrected using the full correction data set.

The "scale only corrected" values show that the torquers have corrupted the magnetometer measurements. Without the correction values, the magnetometer measurements can't be used to measure the Earth's magnetic field. The "corrected" values, however, follow the "commanded" values much more closely and, while there is a small amount of error visible, appear to be usable for measuring the Earth's magnetic field.

Figure 6.5 shows the error for the torquer compensation test. The RMS error for the compensated data is 28 mGauss. The maximum error is 60 mGauss. The RMS error is close to the 26 mGauss to get ±5° error but the maximum error of 60 mGauss is much higher. This is likely due to, in part, to the Y+ SPB being in a location that gives a higher swing than any other board as shown in table 6.1.

6.2.4 Compensation Data Transfer

The compensation data set is generated in MATLAB, but to be used in flight it must be stored on the ACDS board. The calibration data is stored in the internal flash memory on the ACDS microcontroller. The on-board flash has the benefit of reading the same as RAM and is also more abundant but is a bit disruptive to write. This works well in the case of the compensation data because the values are expected to be written once but read often. The other option would be to store the data on the SD card that is used to log data. In this case the calibration data would need

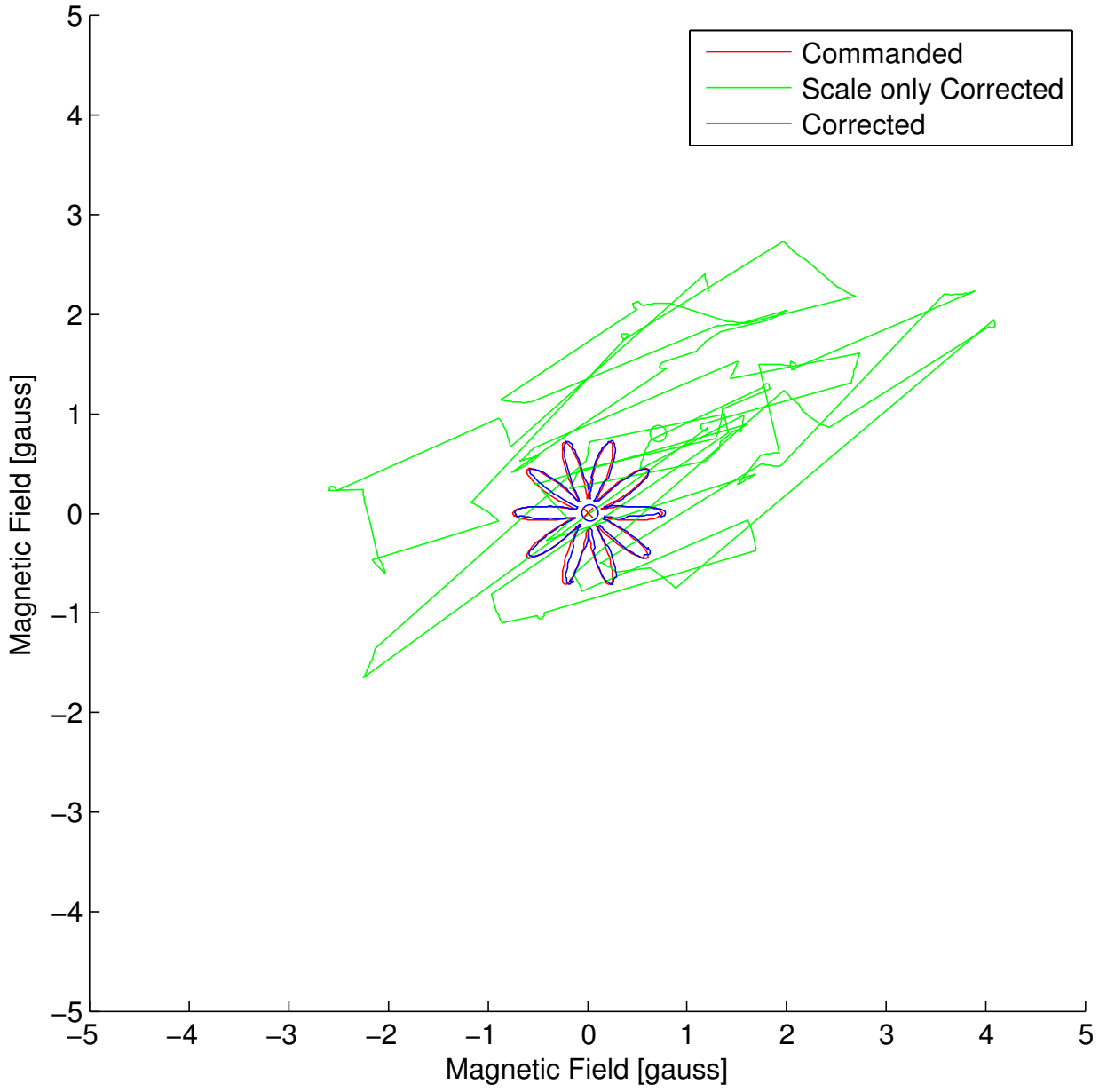


Figure 6.4. Y+ torquer compensation test showing that the correction provides a large amount of improvement over the uncorrected values

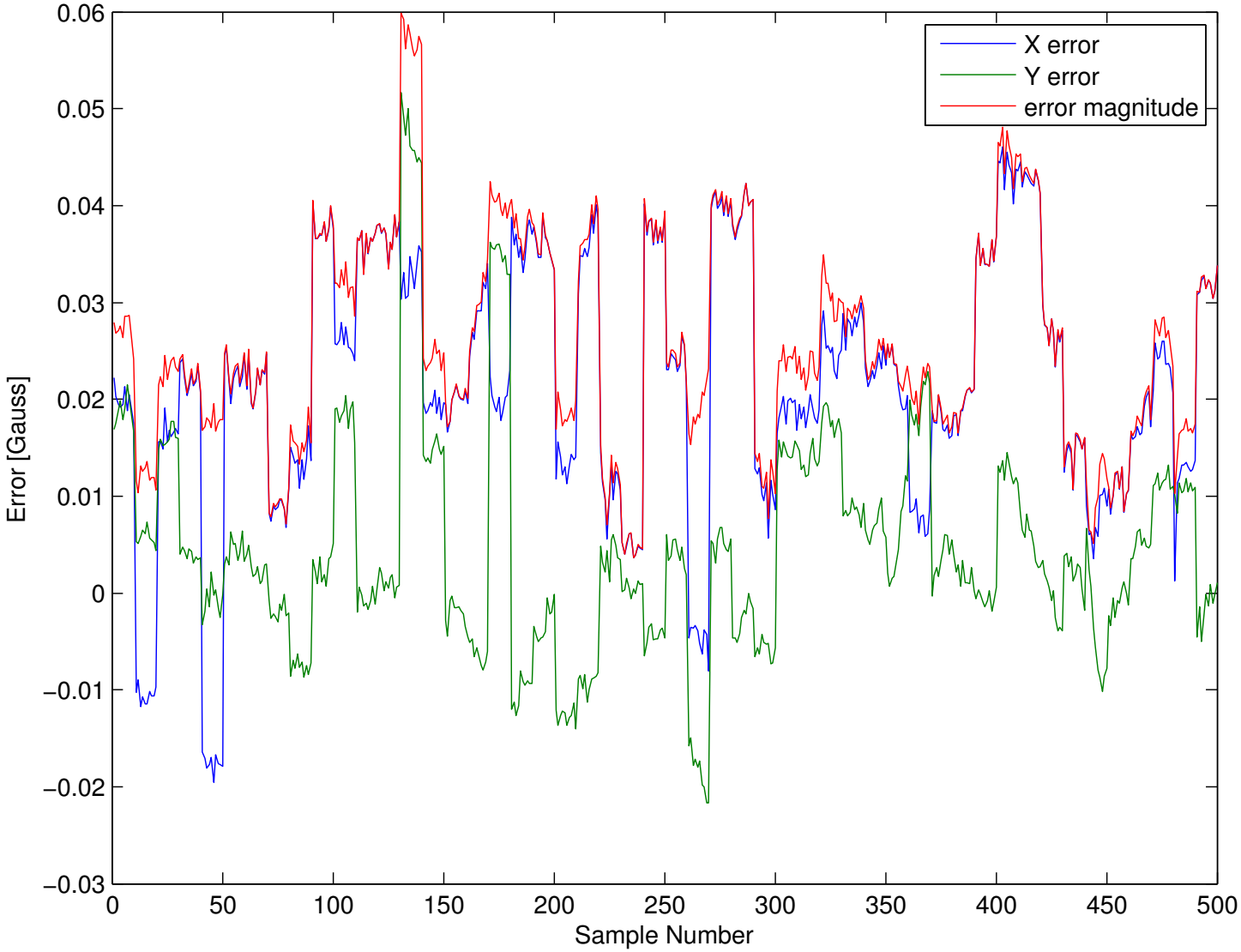


Figure 6.5. Graph showing Y+ torquer compensation error plot

to be buffered in RAM which is much more limited than the flash.

The main flash memory is divided into sectors that are 512 bytes long. Erasing is done on a full sector so the correction data for each SPB is stored in its own sector. The correction data is 408 bytes long. It is stored with header and Cyclic Redundancy Check (CRC) that are 4 bytes long. This leaves 100 extra unused bytes. This is not significant as there is an abundance of flash memory.

Before the correction data is sent to the ACDS it must be repackaged. This is done using the `make_cor_dat` function. Correction data is transfered as single precision IEEE 754 floating point numbers. Because both the ACDS processor and most of the machines commonly used to run MATLAB are little endian machines, the byte order for the transfered data is little endian. It is important to note that in both ends of the transfer no effort is made to make sure that the data order is correct, so this fact would need to be taken into account if the endianness of either the ACDS processor or the lab machine changes.

The data generated with `make_cor_dat` is temporally stored in the SD card on the ACDS board. An identifier is stored with the correction data to identify the corresponding SPB. On the ACDS board the data is unpacked from the SD card, reformatted slightly, and stored in the on-board flash.

The `store_all_cal` function simplifies the process of generating a calibration set. The `store_all_cal` function generates correction data for all SPB boards, sends the data to the ACDS and tells the ACDS to store the data in flash.

6.2.5 Embedded Compensation Testing

The test in figures 6.4 and 6.5 only used the ACDS embedded system to flip torquers and take measurements. MATLAB was used to do all of the calculations for the calibration. During the flight the sensor data must be calibrated using the embedded system on the ACDS board. Furthermore the tests in figures 6.4 and 6.5 were for a single SPB and could only measure two axes. In flight the measurements from all SPBs will be averaged together to get a 3-axis measurement of the magnetic field.

The `tCalTstMSP_all` function is used to test the embedded calibration. The engineering development prototype of the ARC, with compensation data stored on the ACDS processor, is placed in the Helmholtz cage. The `tCalTstMSP_all` function operates in much the same way that the `tCalTstFull`, explained in section 6.2.3, does. The magnetic field is swept through a field sequence. At every point in the sequence the magnetic field is measured and after every 10 points a random torquer is flipped in each axis. The calibrated total magnetic field values as well as the calibrated values from each SPB are read by MATLAB and the results are plotted as shown in

figures 6.6 to 6.8.

Figure 6.6 shows the torquer calibration test using the Y+, Y- and Z+ SPBs with the compensation performed using the embedded system on the ACDS board. The RMS error was 14.6 mGauss. The maximum error was 39.1 mGauss.

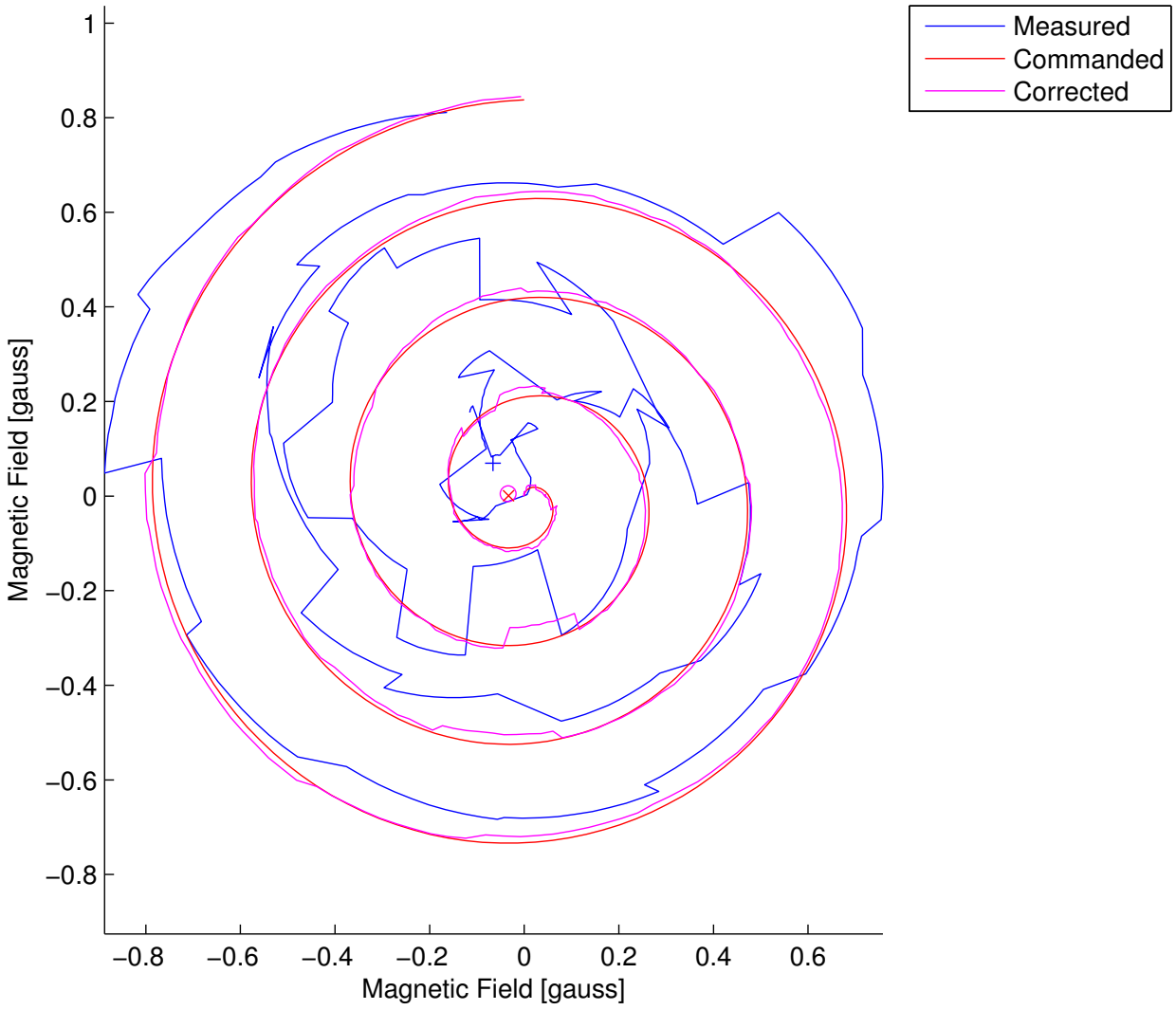


Figure 6.6. Test of the calibration as performed by the ACDS board

Figure 6.7 shows the errors for each sample as well as the torquer states during the samples. The large jumps in the errors all correspond with torquer flips, which indicates that the jumps are due to variations in the torquer offsets and not in the outside magnetic field.

Table 6.2 shows the RMS errors in table form. The N/As in the table are because each SPB only measures in two axes. For each axis the total error is less than the lowest error from a single axis. This indicates that having multiple measurements of the magnetic field reduces the overall error.

Table 6.2. RMS errors in Gauss for ACDS system calibration test

	X-axis	Y-axis	Z-axis
Y+	0.018473	N/A	0.011453
Y-	0.007671	N/A	0.009875
Z+	0.013049	0.010704	N/A
Total	0.007594	0.010704	0.006483

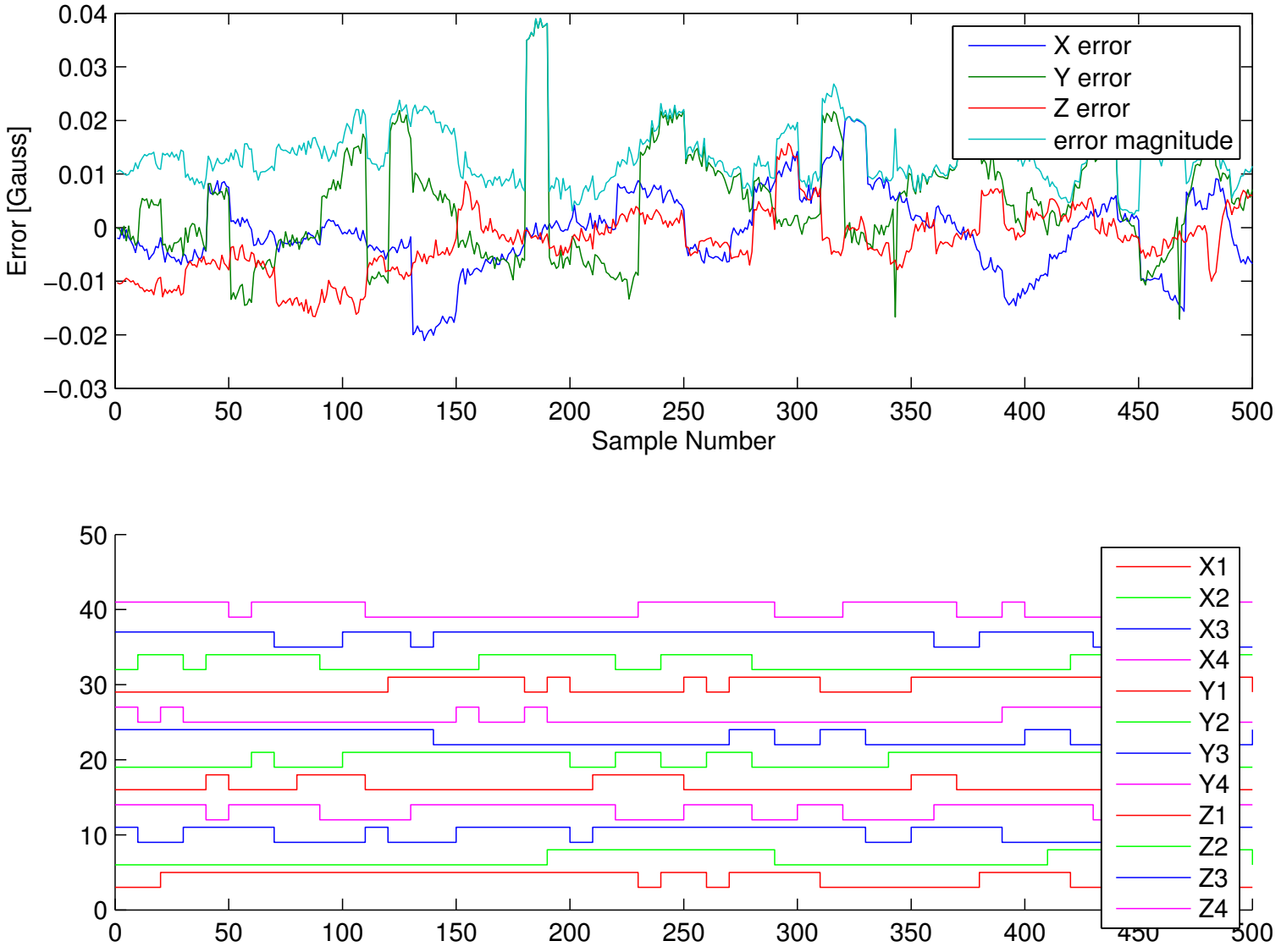


Figure 6.7. Error plot for figure 6.6 showing torquer states

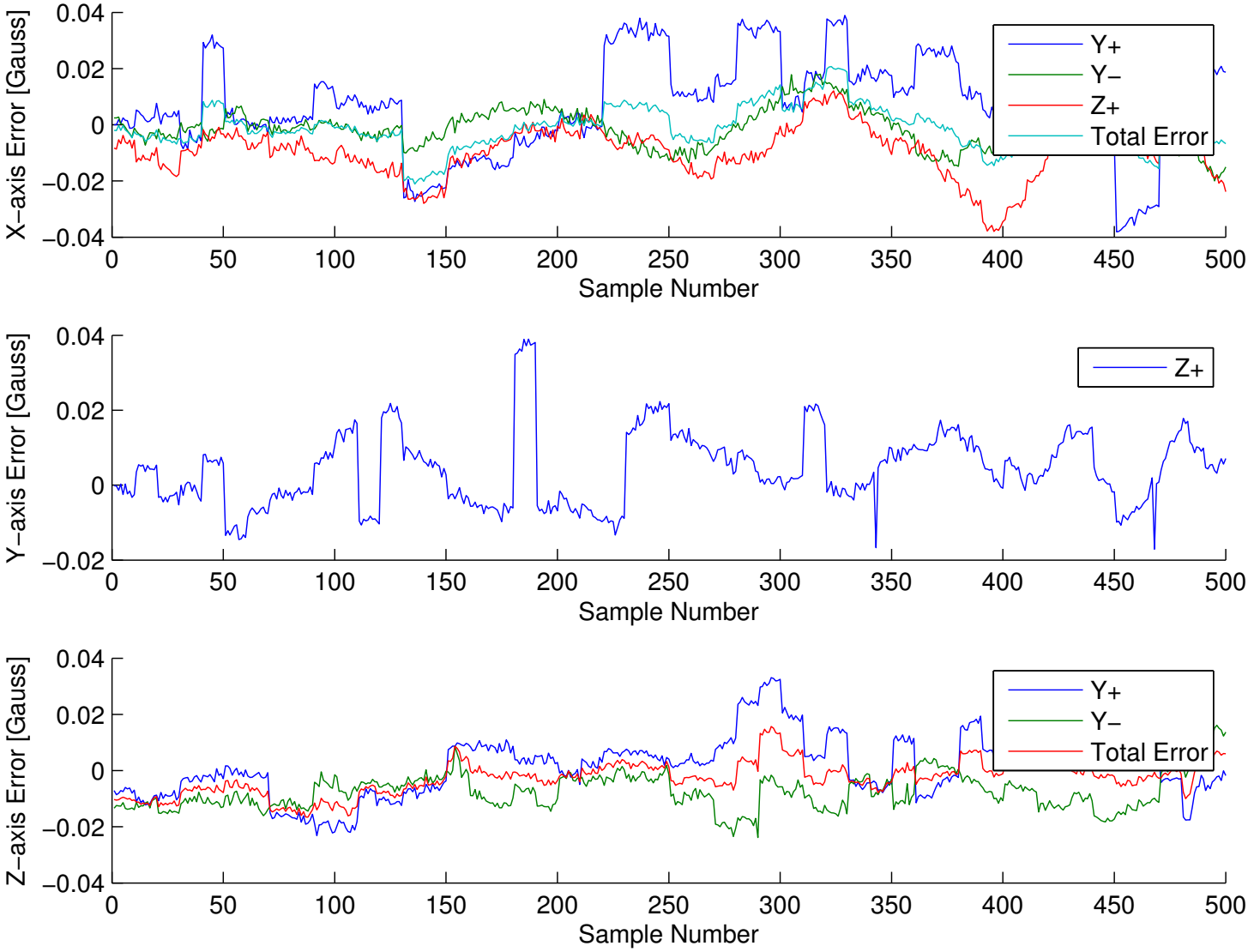


Figure 6.8. Plot of magnetic field errors for each axis

Figure 6.8 shows the errors from each board in each axis.

6.3 B-dot Controller Simulations

To verify the B-dot algorithm, the ACDS is placed in the Helmholtz cage and a rotating field is generated. The minimum rotation rate of the magnetic field that will cause a torquer flip is calculated by starting with equation (6.1)

$$\dot{\vec{B}} = \vec{B} \times \omega \,. \tag{6.1}$$

If the axis of rotation is perpendicular to the field rotation then equation (6.1) becomes equation (6.2)

$$\dot{\vec{B}} = \left|\vec{B}\right| \cdot |\omega| \,. \tag{6.2}$$

The B-dot algorithm equation is shown in equation (6.3),

$$M_{cmd} = C\dot{\vec{B}} \,. \tag{6.3}$$

Substituting equation (6.2) into equation (6.3) and solving for ω gives equation (6.4), where M_{th} is the torque switching threshold for the torquers and ω_{th} is the minimum rotation rate required to flip a torquer,

$$\omega_{th} = \frac{M_{th}}{C\left|\vec{B}\right|} \,. \tag{6.4}$$

The torque switching threshold (M_{th}) is $0.022\,\mathrm{A}\cdot\mathrm{m}^2$, the same as the dipole moment of a single torquer. This can be seen in figure 2.9. The gain, C is empirically determined by simulation and testing to be -4. The magnitude of the Earth's magnetic field, $\left|\vec{B}\right|$, will vary from 0.3 to 0.6 Gauss [18, pp. 114]. A value of 0.3 Gauss gives the worst case (largest) switching threshold so it is used for $\left|\vec{B}\right|$. Equation (6.5) shows the calculation of ω_{th},

$$\omega_{th} = \frac{0.022}{4 \cdot 0.3} = 0.0183 \ \text{rad/sec} \,. \tag{6.5}$$

To understand what the response of the ACDS to a rotating magnetic field should be, a simulation was written. The simulation simply generates a magnetic field sequence and uses the B-dot algorithm to determine what the magnetic torque should be. The simulation quantizes the output torque but does not enforce the one flip per time step requirement.

Figure 6.9 shows a simulation of the detumble algorithm in a spinning magnetic field. The field vector spins about the Z-axis and has a magnitude of 0.3 Gauss. The gain of the simulated controller was -4. The plots show the magnetic dipole moment calculated by the algorithm and the magnetic dipole moment that can be generated as the magnetic field spins around.

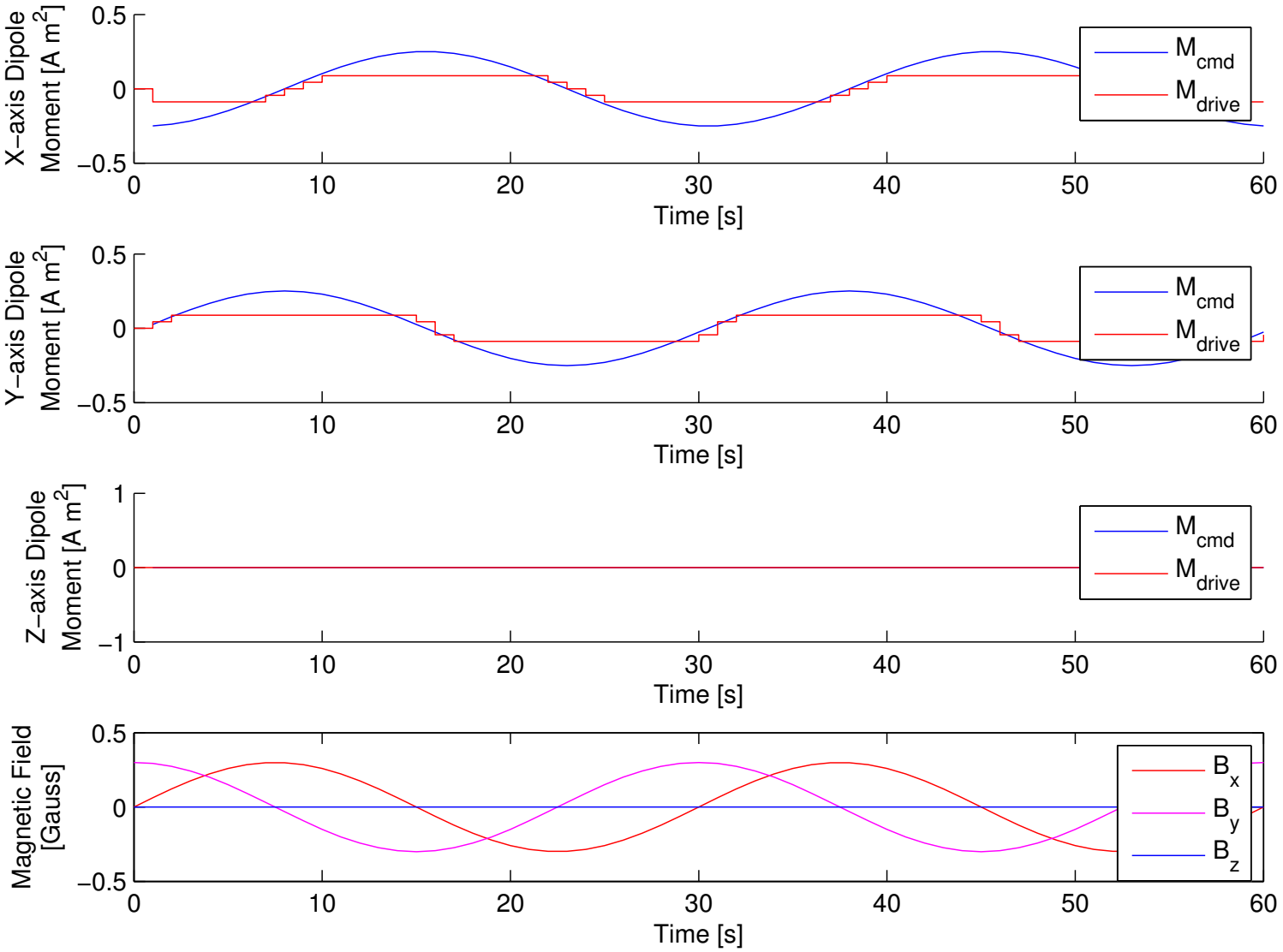


Figure 6.9. Simulation of torquer output with spinning magnetic field

6.4 Detumble Flip Test

To test the detumble algorithm on the ACDS hardware the `bdot_test` function is used. A timer is used to update the magnetic field in the Helmholtz cage every half second. The field is set to a field that by default rotates around the Z-axis at 2 rpm. The ACDS is set to operate in detumble mode and the data that the ACDS outputs are captured.

Figure 6.10 shows the results of the detumble test. This is the same as the detumble simulation in figure 6.9 except that the field has been programed into the Helmholtz cage, measured by the SPBs, calculations performed by the ACDS and torquers flipped in an attempt to stop the motion.

The results in figure 6.10 are similar to the results in the simulation in figure 6.9. The main difference is that there are some unexpected flips in the Z-axis. This is due to noise in the magnetic field causing torquers to be flipped. It is unclear if the noise is caused by the environment in the Helmholtz cage or by torquer flips and imperfect torquer calibration. If the noise is caused by the environment around the Helmholtz cage then the issue can likely be ignored because the magnetic environment is expected to be clean in space. Otherwise the noise can cause undesired flips during flight.

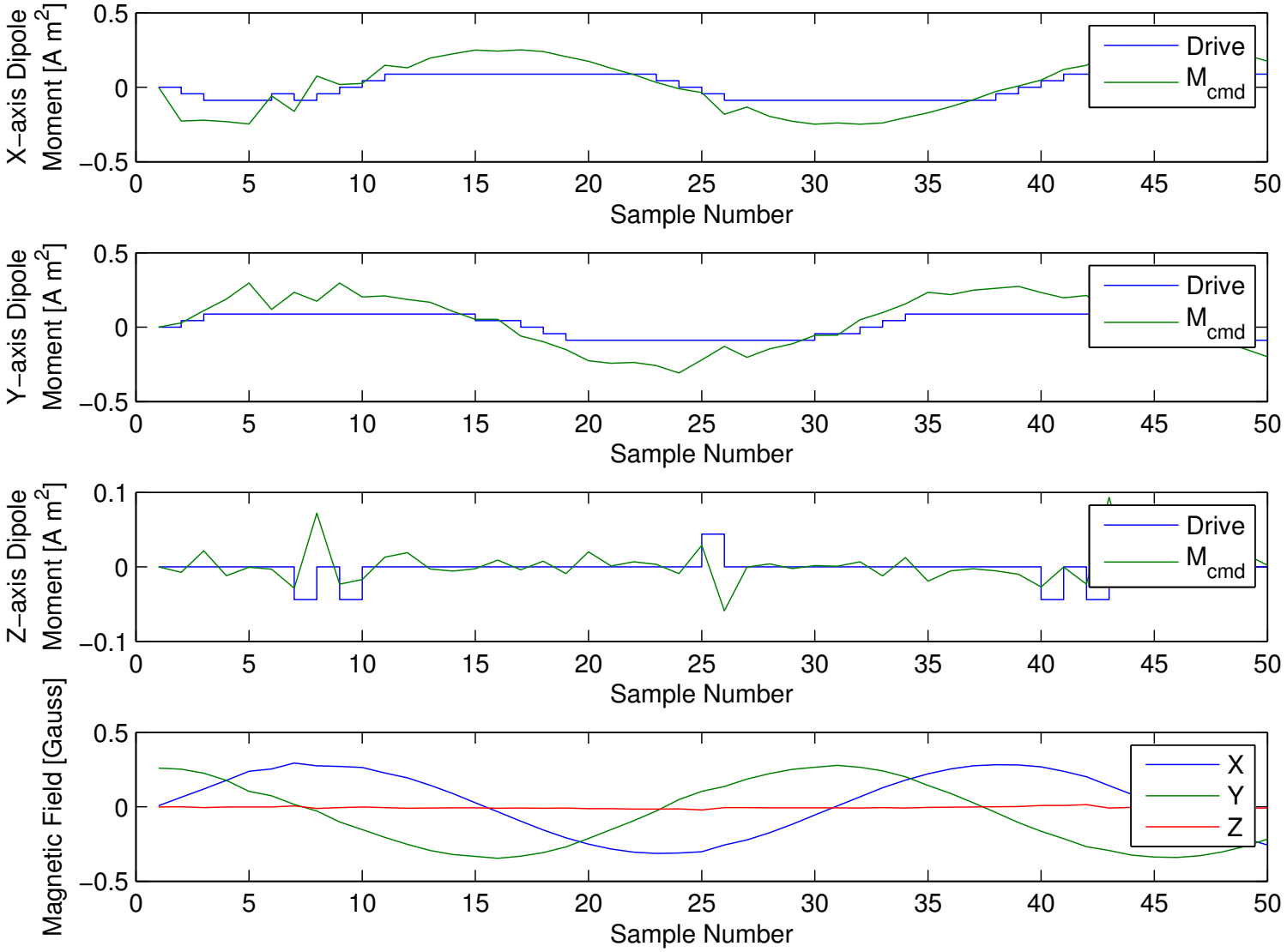


Figure 6.10. Test of the B-dot controller

Chapter 7
Conclusion and Future work

7.1 Conclusion

The goal of this thesis was to make a working prototype of an ACDS using LPMTs, as described by Donald Mentch in [6]. For this thesis flight hardware has been built with software to detumble the satellite. The alignment phase, using bias windows to establish and maintain desired alignment, was not implemented primarily because the rotation rates proved more complex to calculate than originally described in [6]. In addition the required orbit in [6] was circular and for the ARC1 mission the orbit will be elliptical.

In a circular orbit the angular velocity required to remain nadir pointing is constant throughout the orbit. In an elliptical orbit the angular velocity to remain nadir pointing is not constant throughout the orbit. The changing angular velocity presents a problem because the algorithm in [6] works by setting the angular velocity of the satellite to be constant.

A large part of the work for this thesis was focused on the torquer compensation. Torquer compensation is needed so that the Earth's magnetic field can be measured using magnetometers that are not separated from the torquers. It was suggested that torquer compensation was possible but it was not mentioned in [6]. Software has been written to interface with the LEDL board to read magnetometer data and compensate for the torquer field. The torquer compensation has been tested and found to accurately measure the external magnetic field well enough to compute the attitude to within ±5°.

7.2 Future Work

The system built for this thesis does not fully implement everything described in [6]. The main shortcomings of the described system is the software. The hardware is essentially the same as described in [6] except for the torquer sizing.

The torquers as described in [6] are not feasible. The torquer flips in figure 3.10 showed about a ±0.5% variation across multiple flips. The vernier torquers described in [6] were $1/200^{th}$, or 0.5%, the strength of the primary torquers. The result is that a set of primary torquers that is set to produce zero torque could produce the same amount of torque as a pair of vernier torquers. The solution used in this thesis was to not use the vernier torquers and fill the empty slots with primary torquers. This could be improved upon if a torquer core could be made with approximately 10% the dipole moment of the alnico torquers.

The rotation rate determination algorithm was found to be a much more complex problem than described in [6] which suggested that rotation rates could be determined directly from magnetic field measurements. Upon further investigation it was found that this was not feasible for the

alignment phase. As an alternative the rotation rates can be determined from the magnetic field using a Kalman filter. The Kalman filter brings an additional level of complexity that the original system attempted to avoid.

The Kalman filter uses a magnetic field model for attitude determination. The magnetic field model uses the location of the satellite to calculate the expected local magnetic field in the orbital reference frame. This requires that the position of the satellite be known for the Kalman filter to work. The location could be calculated by uplinking the orbital parameters of the satellite and using an orbit propagator to calculate position as the satellite moves through its orbit.

The bias windows were originally supposed to be determined using the magnetic field to estimate the latitude. With the Kalman filter the location can be directly determined using the orbit propagator so there is no need for latitude determination from the Earth's magnetic field.

7.2.1 New Concept of Operations

The algorithm shown in figure 5.2 with the above additions is shown in figure 7.1. The orbital parameters are used to determine the bias window and to determine the local magnetic field direction. The Kalman filter is used to output rates to the torque algorithm. The bias window is used by the torque algorithm to determine when and how to bias the torquers.

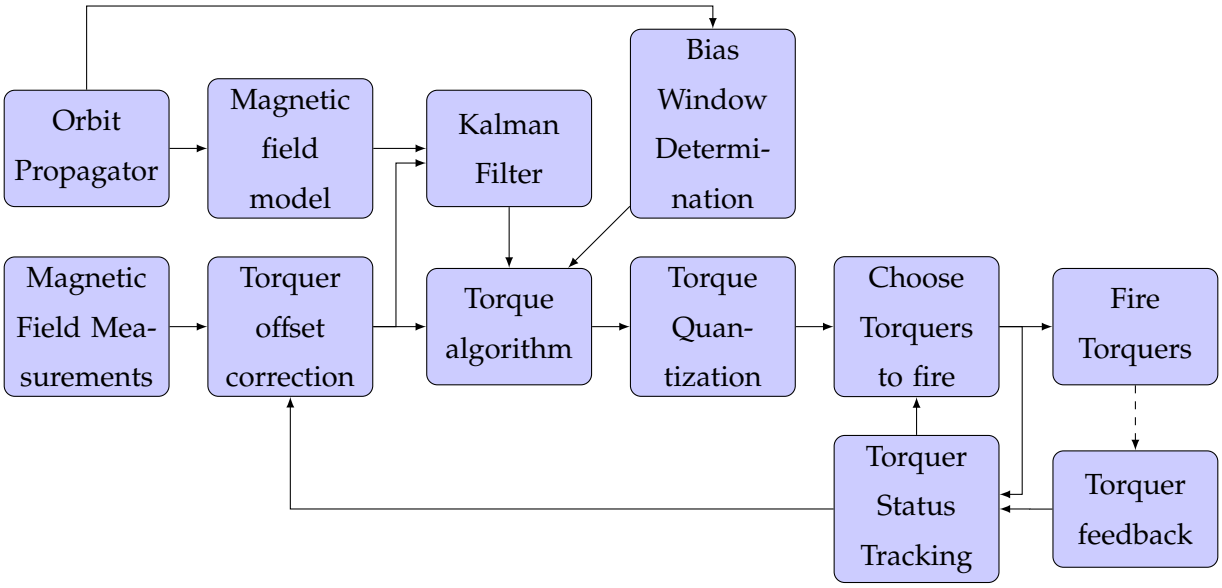


Figure 7.1. Overall Software Block Diagram

The torque algorithm varies depending on the current mode. In mode 1 the Kalman filter and bias window determination are not used and the algorithm is the one showin in figure 5.2. In modes 2 and 3 the algorithms shown in figures 7.2 and 7.3 are used.

7.2.1.1 Mode 2

Figure 7.2 shows the Mode 2 torque algorithm block diagram. Torque is calculated the same way as in Mode 1 but this time a bias is added depending on which region of the orbit the satellite is in. The bias tends to cause the satellite to rotate. This causes the algorithm to cancel out the bias. This is prevented by preventing the resulting dipole moment from being in the opposite direction as the bias. Outside of the bias regions the torquers are set to produce no torque.

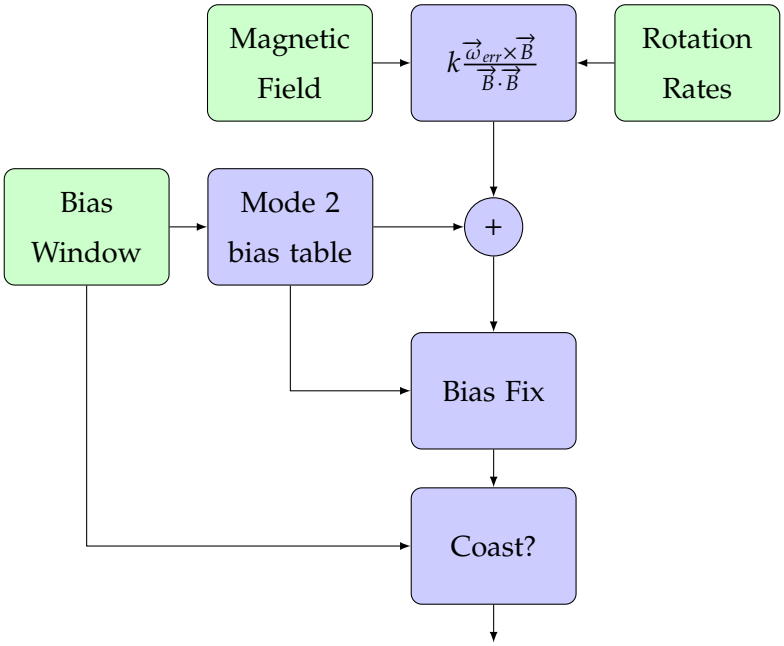


Figure 7.2. Mode 2 Torque Algorithm Block Diagram

7.2.1.2 Mode 3

Figure 7.3 shows the Mode 3 torque algorithm block diagram. This is similar to Mode 2 except only the north pole bias window is used and outside the bias window torque is generated to slow rotation rates.

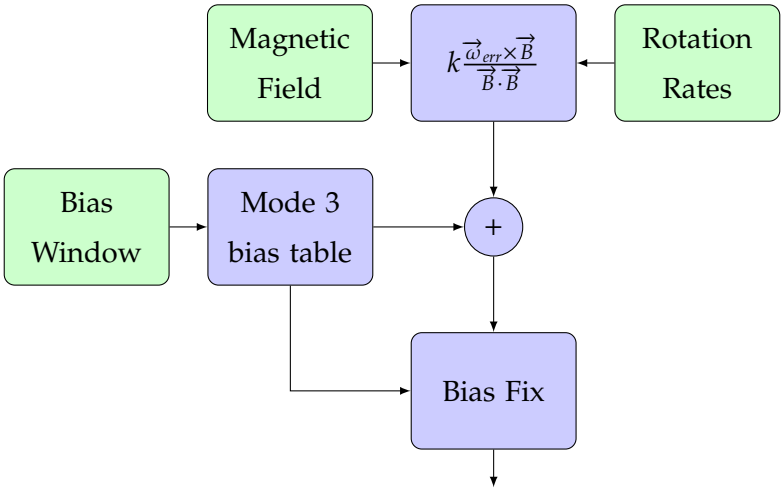


Figure 7.3. Mode 3 Torque Algorithm Block Diagram

7.2.2 Alternate Operations

The original system presented in [6] used the biasing scheme to get the satellite into the correct attitude without ever knowing the attitude. The Kalman filter proposed for rate determination must compute attitude in order to compute the rotation rates. Because the attitude is known a more traditional attitude control algorithm to be used. Using a more traditional algorithm will allow the ACDS to function in orbits that are elliptical.

References

[1] (2014, Aug.) Cubesat design specification. [Online]. Available: http://www.cubesat.org/images/developers/cds_rev13_final.pdf

[2] (2014, Aug.) CubeSat-based Science Missions for Geospace and Atmospheric Research. [Online]. Available: http://www.nsf.gov/pubs/2014/nsf14535/nsf14535.pdf

[3] A. Ali, L. Reyneri, J. de los Rios, H. Ali, and M. Mughal, "Reconfigurable magnetorquer for the cubepmt module of cubesat satellites," in *Multitopic Conference (INMIC), 2012 15th International*, Dec 2012, pp. 178–183.

[4] Clyde Space, 2012, small Sattelite Solar Pannels. [Online]. Available: http://www.clyde-space.com/documents/2625

[5] Alaska Space Grant Program, November 2013. [Online]. Available: http://spacegrant.alaska.edu/content/student-satellite-program

[6] D. Mentch, "Picosatellite attitude control utilizing magnetometers & low-power magnetic torquers," Ph.D. dissertation, University of Alaska Fairbanks, 2011.

[7] M. E. Polites, "A low-power magnetic torquer for satellite attitude control," Master's thesis, University of Alabama Tuscaloosa, Tuscaloosa, Alabama, 1971.

[8] E. J. Sturm II, "Magnetic attitude estimation of a tumbling spacecraft," Master's thesis, California Polytechnic State University, July 2005.

[9] F. B. Vej 7, "Attitude Determination and Control System for AAUSAT3," Master's thesis, Aalborg University, Jun 2010.

[10] Clyde Space, 2012, cubeSat Electrical Power Systems. [Online]. Available: http://www.clyde-space.com/cubesat_shop/eps

[11] (2013, Jun.) Cubesat kit websight. [Online]. Available: http://www.cubesatkit.com/index.html

[12] May 2014, magnetic Properties of Alnico Materials. [Online]. Available: http://www.advancedmagnetsource.com/mmpa0100-alnicomat.pdf

[13] Dexter magnetic technologies, May 2014, field on Axis of a Cylindrical Magnet Calculator. [Online]. Available: http://www.dextermag.com/resource-center/magnetic-field-calculators/field-on-axis-of-cylindrical-magnet-calculator

[14] Honeywell, 2012, Appl. Note AN215. [Online]. Available: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/Magnetic__Literature_Application_notes-documents/AN215_Cross_Axis_Effect_for_AMR_Magnetic_Sensors.pdf

[15] ——, 2013, 1, 2 and 3 Axis Magnetic Sensors HMC1051/HMC1052L/HMC1053. [Online]. Available: http://www51.honeywell.com/aero/common/documents/myaerospacecatalog-documents/Defense_Brochures-documents/HMC_1051-1052-1053_Data_Sheet.pdf

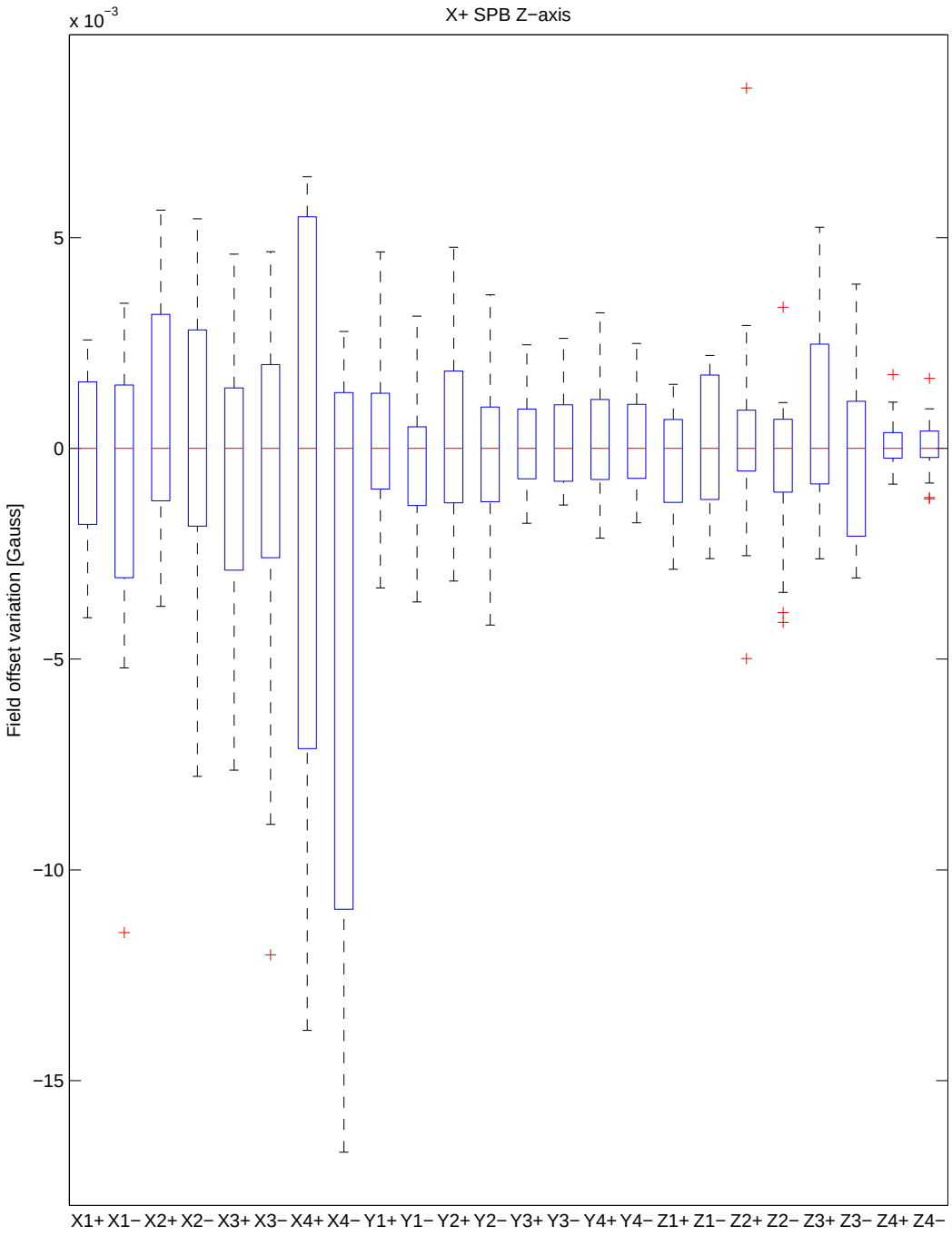
[16] J. Frey, J. Hawkins, and D. Thorsen, "Magnetometer calibration in the presence of hard magnetic torquers," in *Aerospace Conference, 2014 IEEE*, March 2014, pp. 1–6.

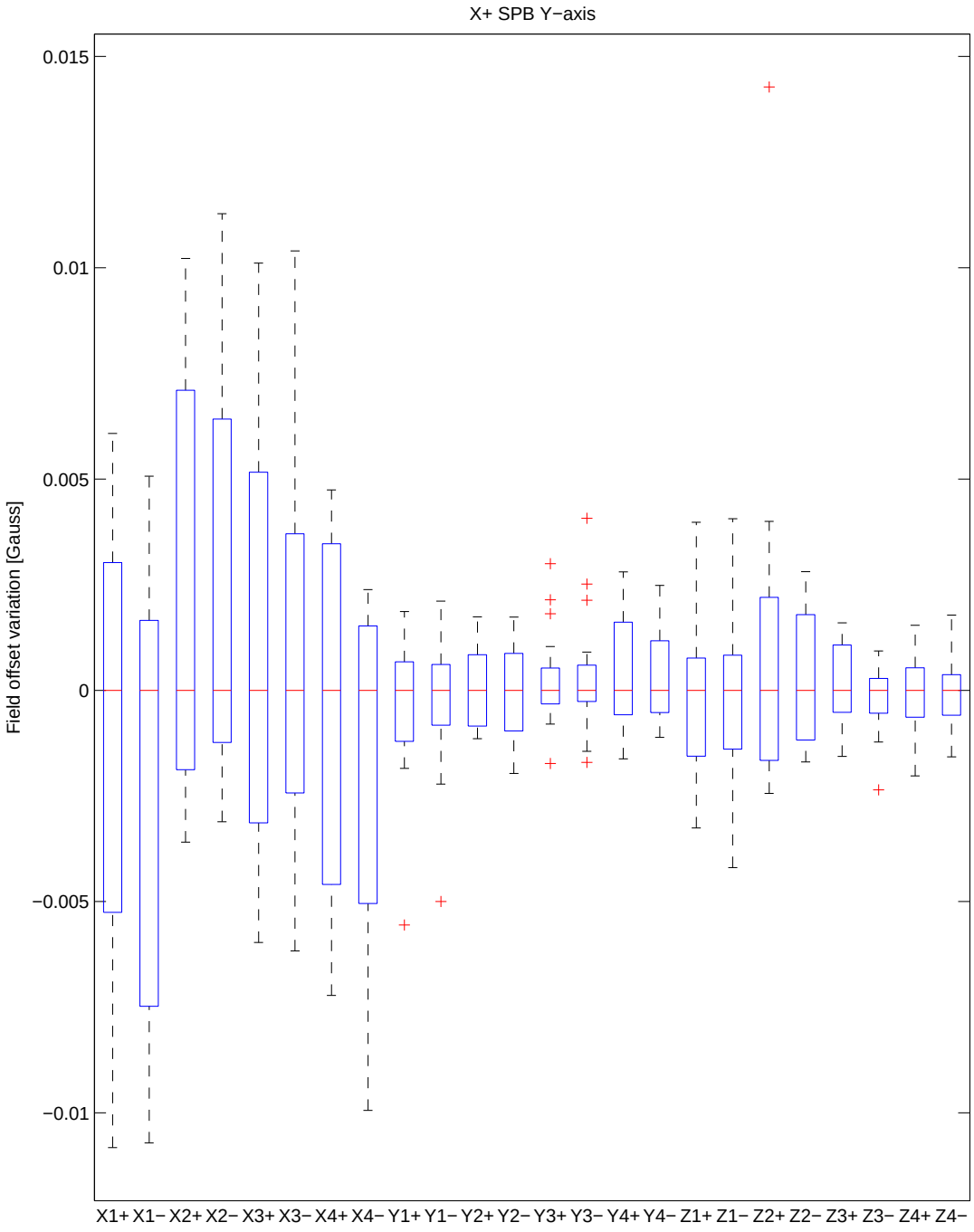
[17] E. Silani and M. Lovera, "Magnetic spacecraft attitude control: a survey and some new results," *Control Engineering Practice*, vol. 13, no. 3, pp. 357 – 371, 2005, aerospace {IFAC} 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066103002922

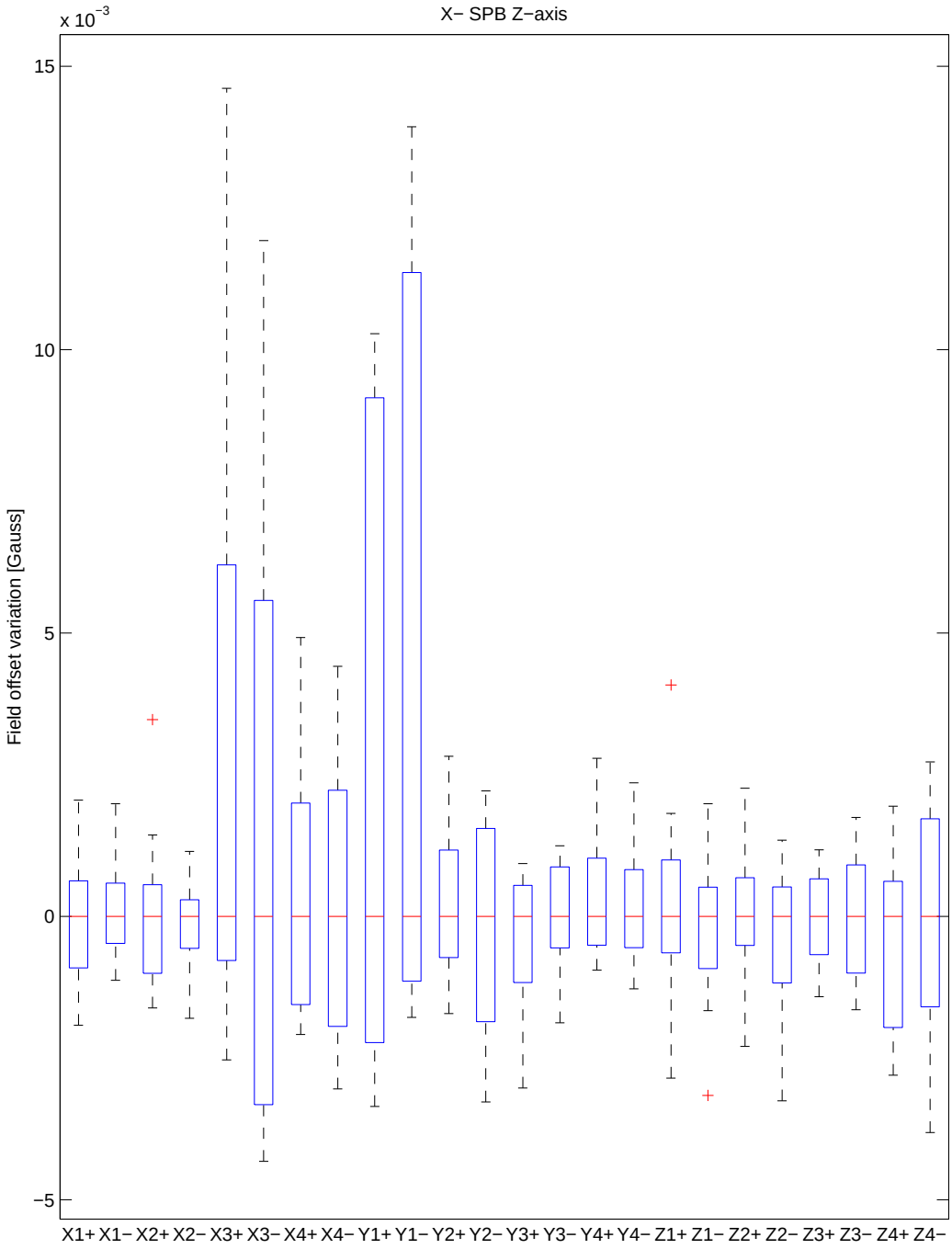
[18] J. R. Wertz, Ed., *Spacecraft Attitude Determination and Control*. Kluwer, 1991.

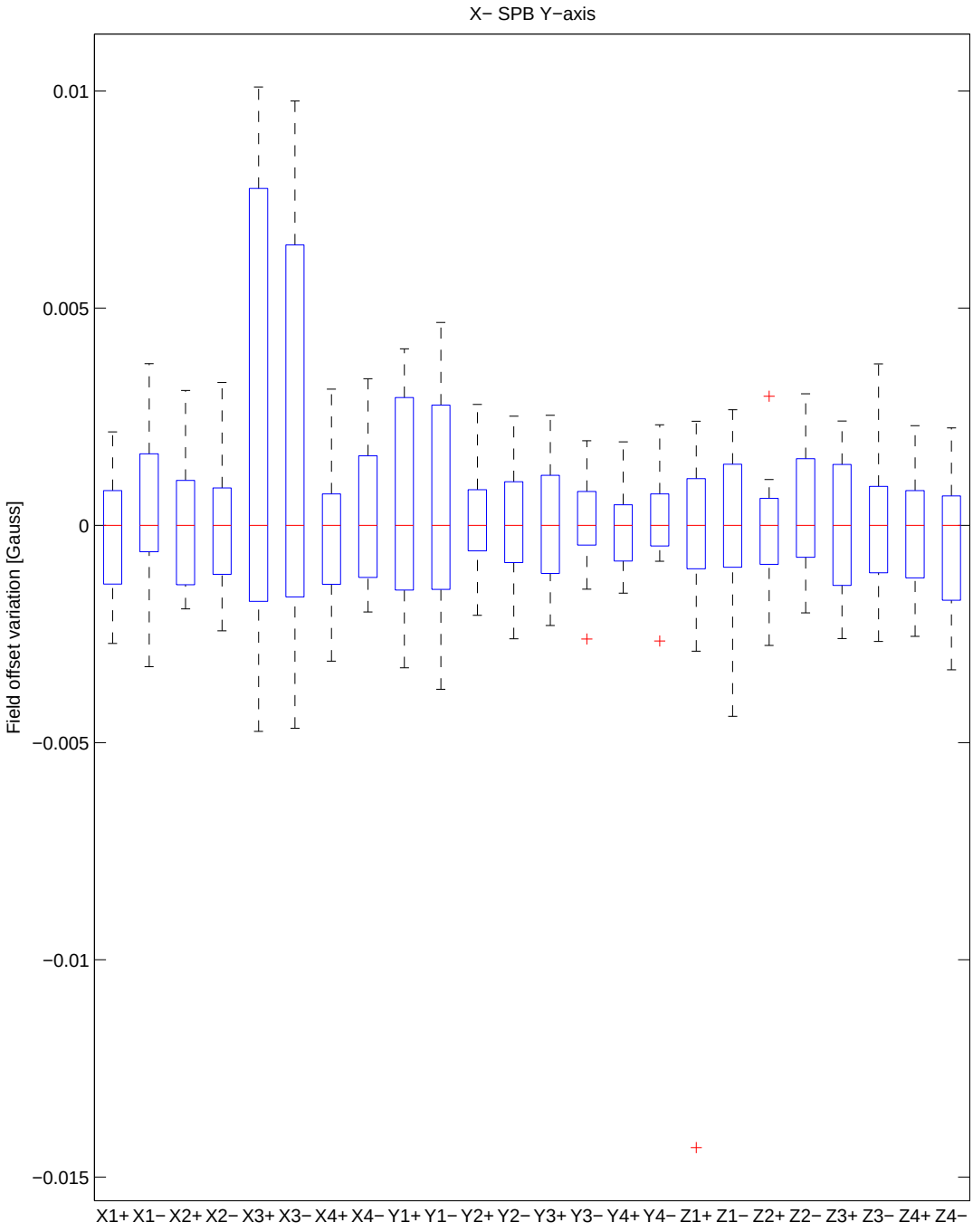
[19] (2014, Aug.) Box plot - MATLAB boxplot. [Online]. Available: http://www.mathworks.com/help/stats/boxplot.html

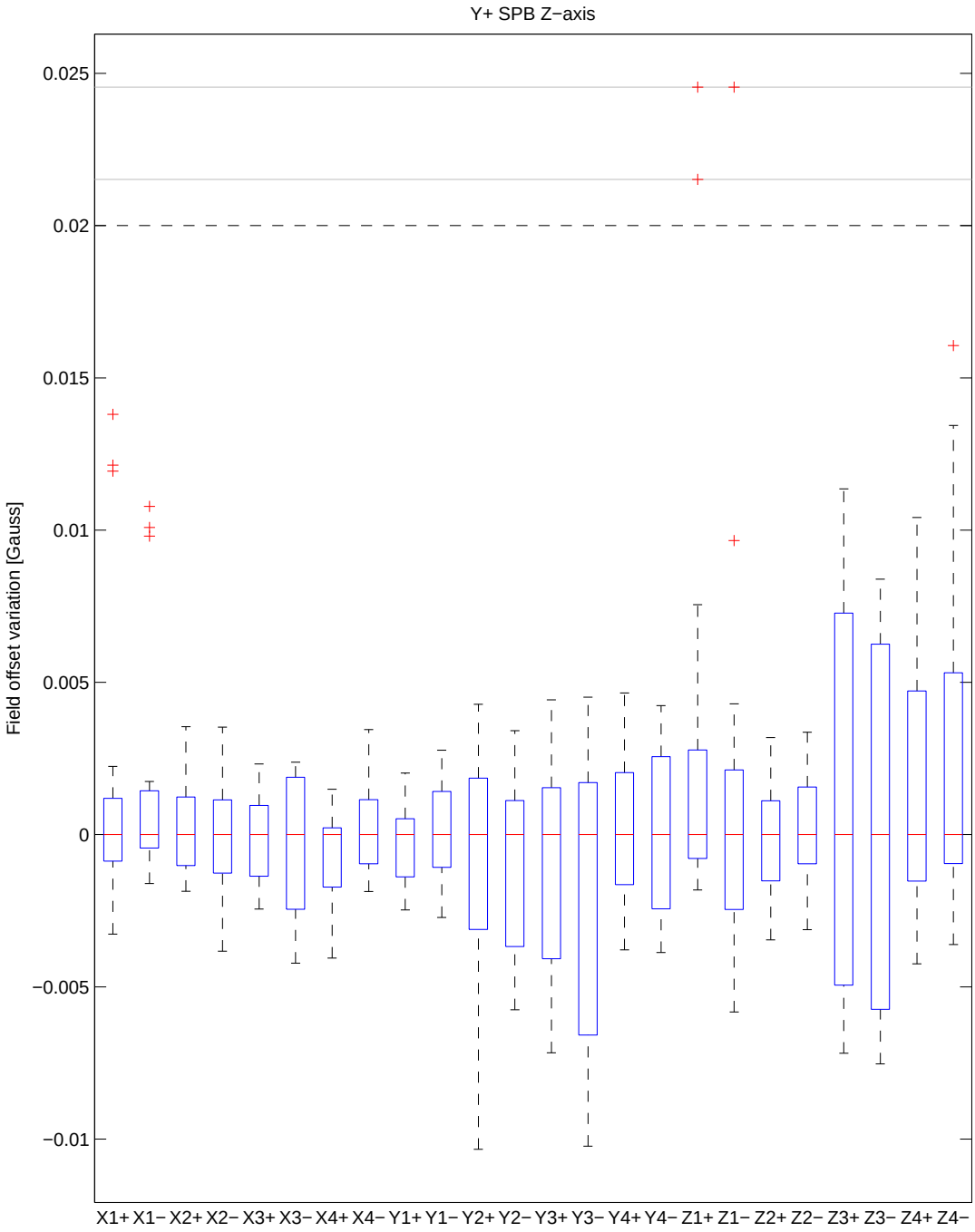
Appendix A
Offset Plots

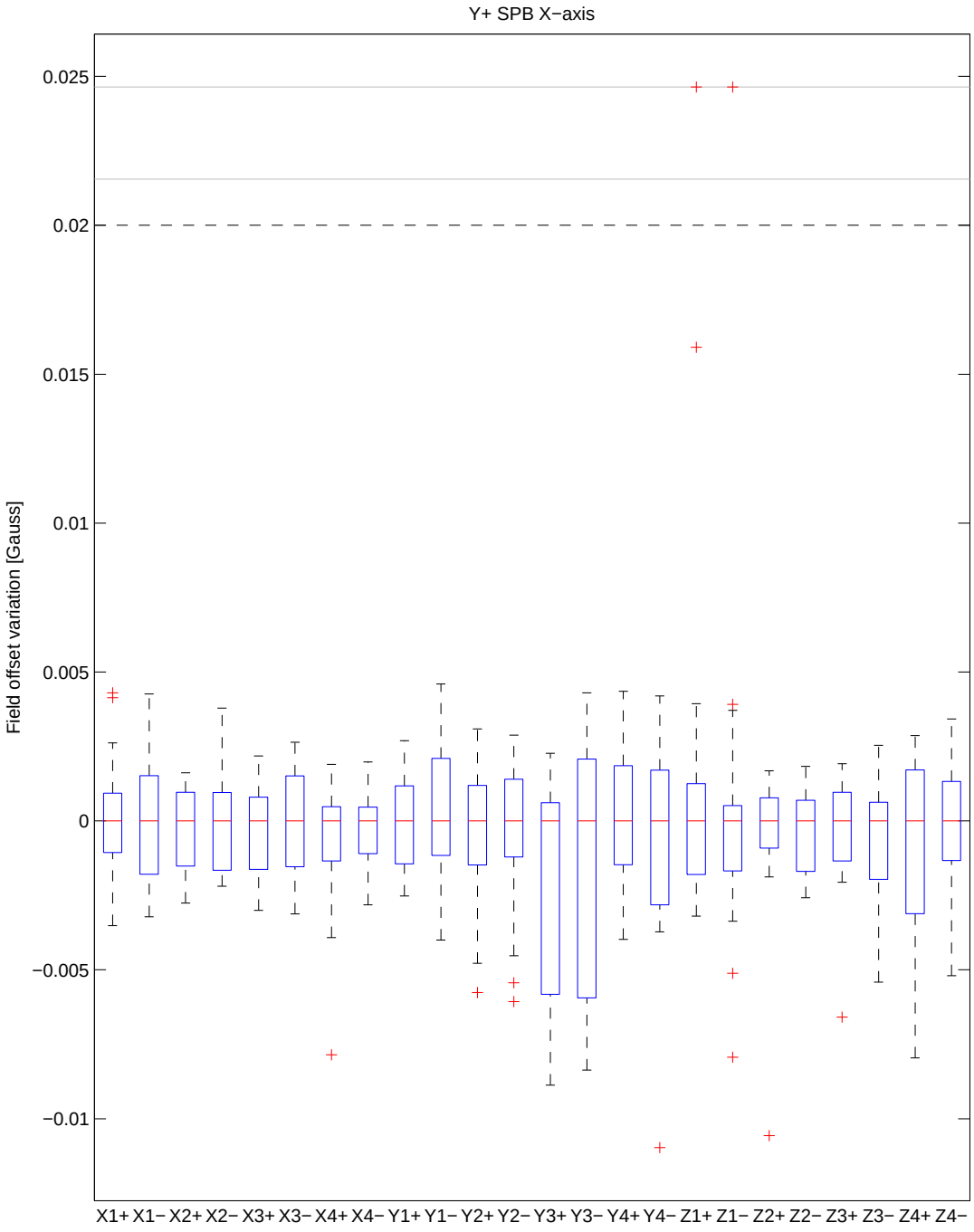


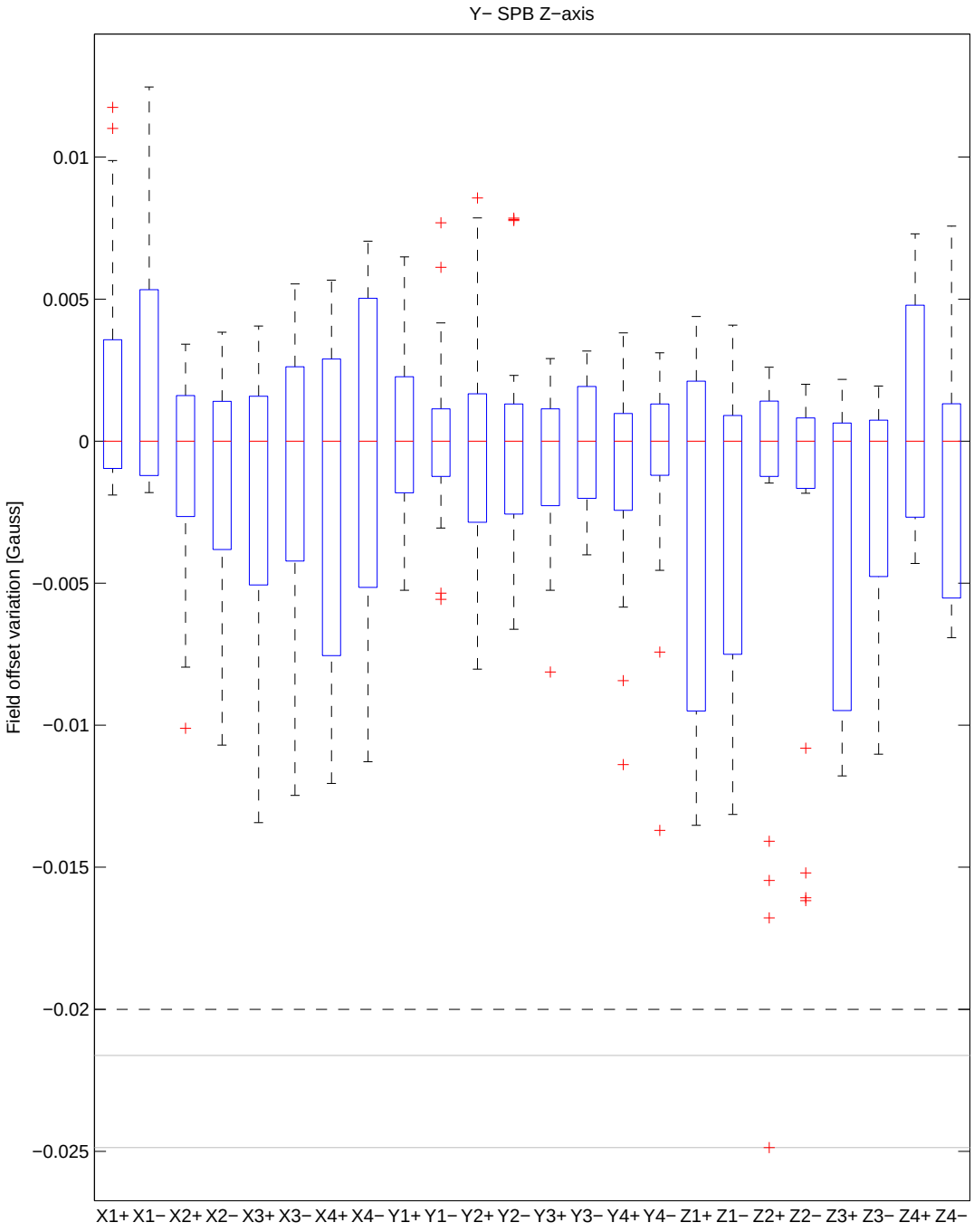

X+ SPB Y−axis
Field offset variation [Gauss]
0.015
0.01
0.005
0
−0.005
−0.01
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−

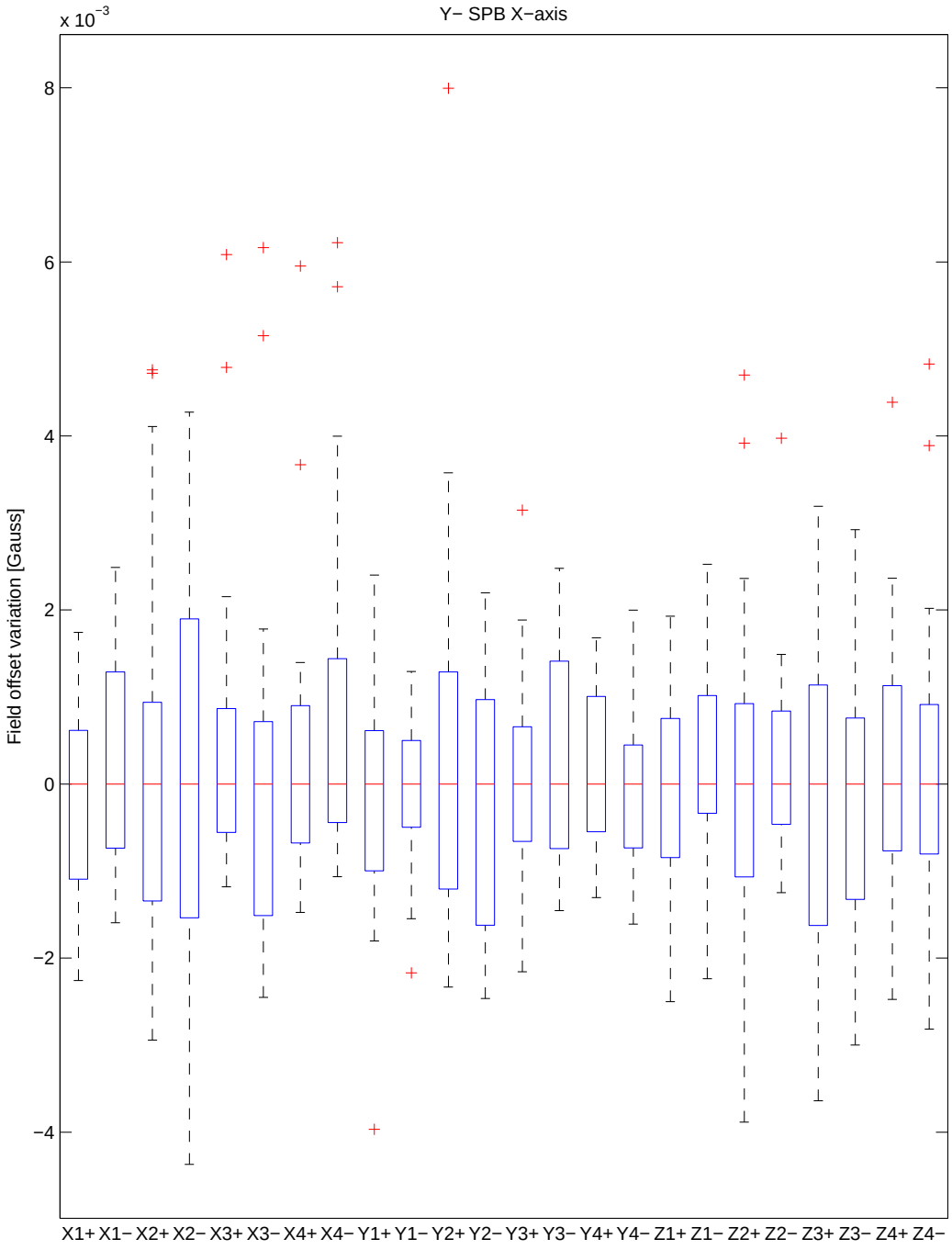

X− SPB Z−axis
x 10^−3
Field offset variation [Gauss]
15
10
5
0
−5
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−

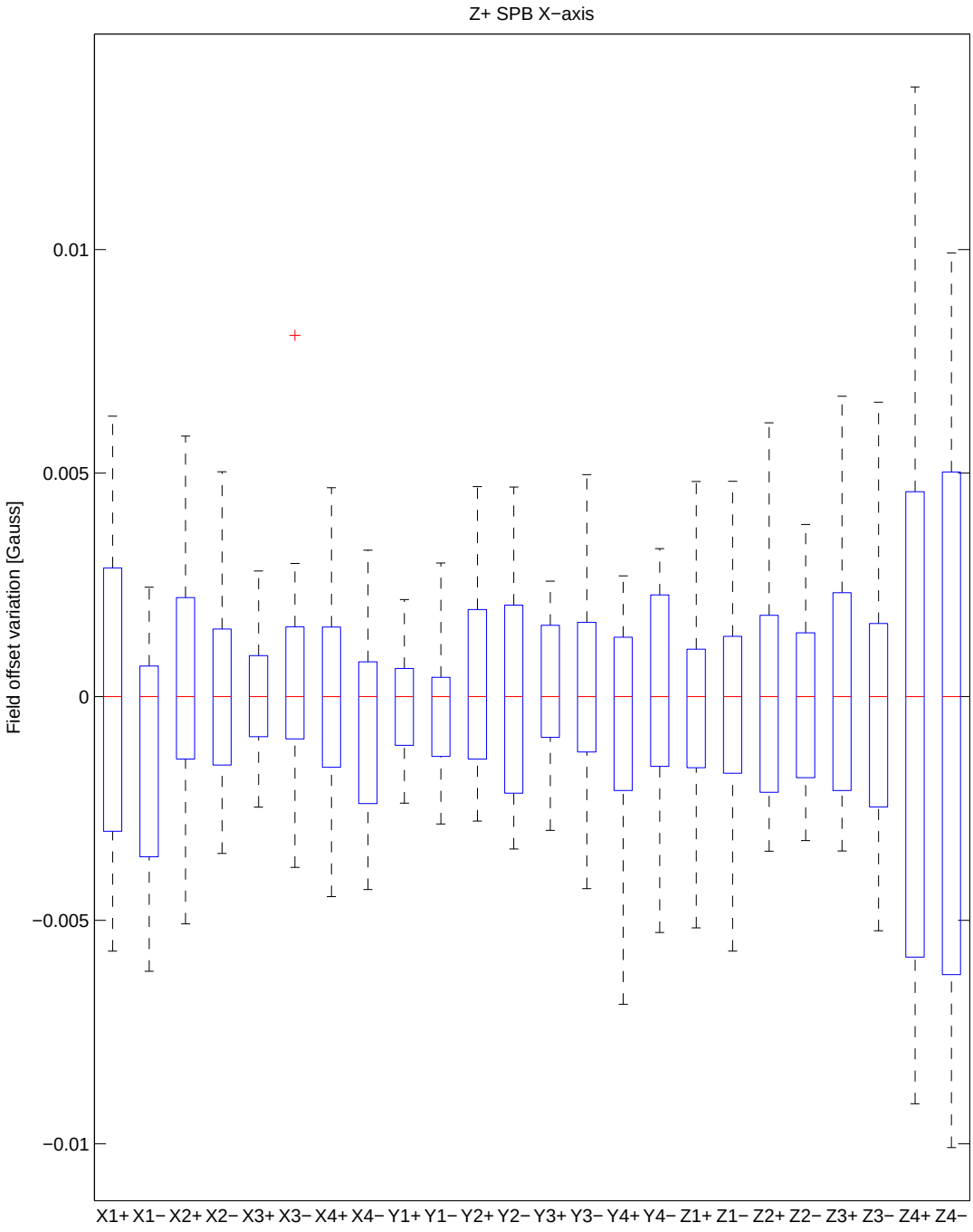

X− SPB Y−axis
Field offset variation [Gauss]
0.01
0.005
0
−0.005
−0.01
−0.015
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−

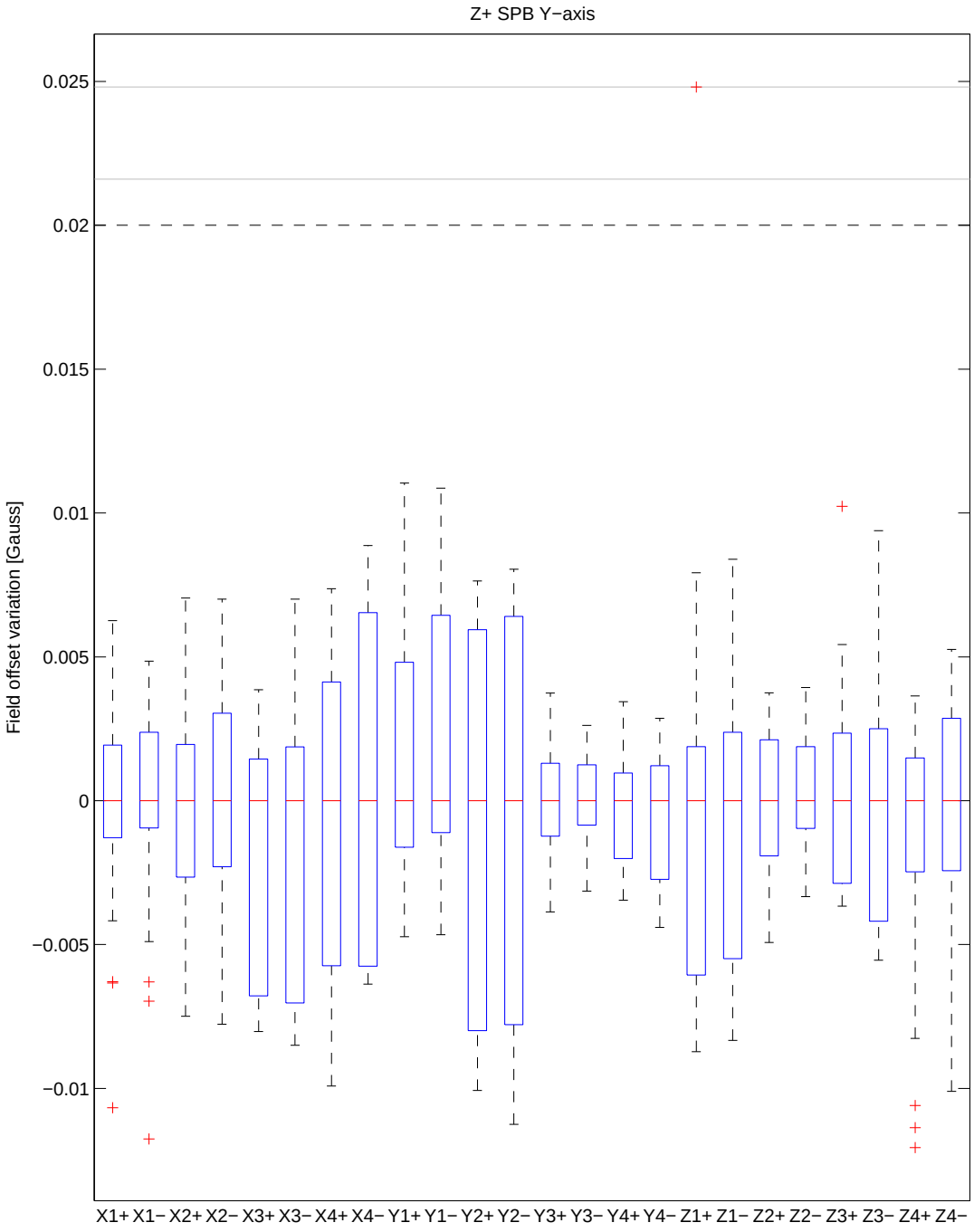

Y+ SPB Z−axis
Field offset variation [Gauss]
0.025
0.02
0.015
0.01
0.005
0
−0.005
−0.01
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−


Y+ SPB X−axis
Field offset variation [Gauss]
0.025
0.02
0.015
0.01
0.005
0
−0.005
−0.01
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−


Y− SPB Z−axis
Field offset variation [Gauss]
0.01
0.005
0
−0.005
−0.01
−0.015
−0.02
−0.025
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−


Y− SPB X−axis
x 10^-3
Field offset variation [Gauss]
8
6
4
2
0
−2
−4
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−


Z+ SPB X−axis
Field offset variation [Gauss]
0.01
0.005
0
−0.005
−0.01
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−


Z+ SPB Y−axis
Field offset variation [Gauss]
0.025
0.02
0.015
0.01
0.005
0
−0.005
−0.01
X1+ X1− X2+ X2− X3+ X3− X4+ X4− Y1+ Y1− Y2+ Y2− Y3+ Y3− Y4+ Y4− Z1+ Z1− Z2+ Z2− Z3+ Z3− Z4+ Z4−

Appendix B
Calibration Equation Derivation

Equation (B.1) shows the equation from [14] as shown in equation (3.3),

$$V_s = V_b \left(S_s H_s + D \cdot H_c + V_{os}\right) . \tag{B.1}$$

The substitution

$$V_s' = \frac{V_s}{V_b} , \tag{B.2}$$

shown in equation (3.4), is made to get

$$V_s' = S_s H_s + D \cdot H_c + V_{os} . \tag{B.3}$$

To solve for H_c and H_s equation (B.3) is duplicated for the cross axis direction to get

$$\begin{aligned} V_s' - V_{oss} &= S_{ss} H_s + D_s \cdot H_c , \\ V' - V_{oss} &= S_{sc} H_c + D_c \cdot H_s . \end{aligned} \tag{B.4}$$

To solve, the equations are put into matrix form as follows:

$$\begin{bmatrix} V_s' - V_{oss} \\ V_c' - V_{osc} \end{bmatrix} = \begin{bmatrix} S_{ss} & D_s \\ D_c & S_{sc} \end{bmatrix} \begin{bmatrix} H_s \\ H_c \end{bmatrix} . \tag{B.5}$$

The equations are solved using the 2X2 matrix inverse to get

$$\begin{bmatrix} H_s \\ H_c \end{bmatrix} = \begin{bmatrix} S_{ss} & D_s \\ D_c & S_{sc} \end{bmatrix}^{-1} \begin{bmatrix} V_s' - V_{oss} \\ V_c' - V_{osc} \end{bmatrix} = \frac{1}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \begin{bmatrix} S_{sc} & -D_s \\ -D_c & S_{ss} \end{bmatrix} \begin{bmatrix} V_s' - V_{oss} \\ V_c' - V_{osc} \end{bmatrix} \tag{B.6}$$

$$\begin{aligned} H_s &= \frac{S_{sc}\left(V_s' - V_{oss}\right)}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} - \frac{D_s\left(V_c' - V_{osc}\right)}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\ H_c &= -\frac{D_c\left(V_s' - V_{oss}\right)}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} + \frac{S_{ss}\left(V_c' - V_{osc}\right)}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} . \end{aligned} \tag{B.7}$$

Equation (B.8) shows equation (B.7) after expanding and gathering like terms,

$$\begin{aligned} H_s &= \frac{S_{sc} \cdot V_s'}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} - \frac{D_s \cdot V_c'}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} - \frac{S_{sc} \cdot V_{oss} - D_s \cdot V_{osc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\ H_c &= -\frac{D_c \cdot V_s'}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} + \frac{S_{sc} \cdot V_c'}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} - \frac{S_{ss} \cdot V_{oss} - D_c \cdot V_{osc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} . \end{aligned} \tag{B.8}$$

Equation (B.8) is simplified using the substitutions,

$$
\begin{aligned}
C_1 &= \frac{S_{sc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\
C_2 &= -\frac{D_s}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\
C_3 &= -\frac{S_{sc} \cdot V_{oss} - D_s \cdot V_{osc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\
C_4 &= -\frac{D_c}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\
C_5 &= \frac{S_{sc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \\
C_6 &= -\frac{S_{ss} \cdot V_{oss} - D_c \cdot V_{osc}}{S_{sc} \cdot S_{ss} - D_s \cdot D_c} \,,
\end{aligned}
\tag{B.9}
$$

to get

$$
\begin{aligned}
H_s &= C_1 \cdot V_s' + C_2 \cdot V_c' + C_3 \\
H_c &= C_4 \cdot V_s' + C_5 \cdot V_c' + C_6 \,,
\end{aligned}
\tag{B.10}
$$

which is the same as equation (3.6).